DM 14,– öS 120,– sfr 14,–

CPC INTERNATIONAL SONDERHEFT

Sonderheft

8-89/90

CP/M: Alles über das Betriebssystem



Spiele:

Let's work
– Super Strategiespiel

Hanseat
– Handel im Mittelalter

Tips & Tricks:

System-Calls
Aufrüstung des DMP 2000
Einzelschrittsimulator

Hardware:

Sprungmatte mit Joystickanschluß

Grundlagen:

Datenfernübertragung
– Das Tor zur Welt

Anwendungen:

Weltzeituhr im PC
Satellitenbahnen berechnen
Fraktalgenerator zum Abtippen

Tolle Anwendungs-programme

SOFTWARE EXPERIMENT

Autor: Matthias Uphoff

Von der Idee zum Programm

Nehmen Sie teil an dem Abenteuer "Computer"! In zehn lebendig und leicht verständlich geschilderten Reisen in die Welt der Computergrafik, Simulation und künstlichen Intelligenz erarbeitet der Autor mit Ihnen zusammen insgesamt 17 interessante Programme, die Sie auf Diskette/Kassette erhalten.

Das umfangreiche, 180 seitige Handbuch vermittelt Ihnen auf anschauliche und unterhaltsame Weise, wie aus Algorithmen Datenstrukturen und letztendlich Software entsteht: Von der Idee zum Programm - der ideale Kursus zum Einstieg in das Computerwissen, für Hobby und Schulunterricht. Sehen Sie selbst, zu welchen phantastischen Grafiken und verblüffenden Intelligenzleistungen der CPC bei raffinierter Programmierung fähig ist!

Weiterhin:

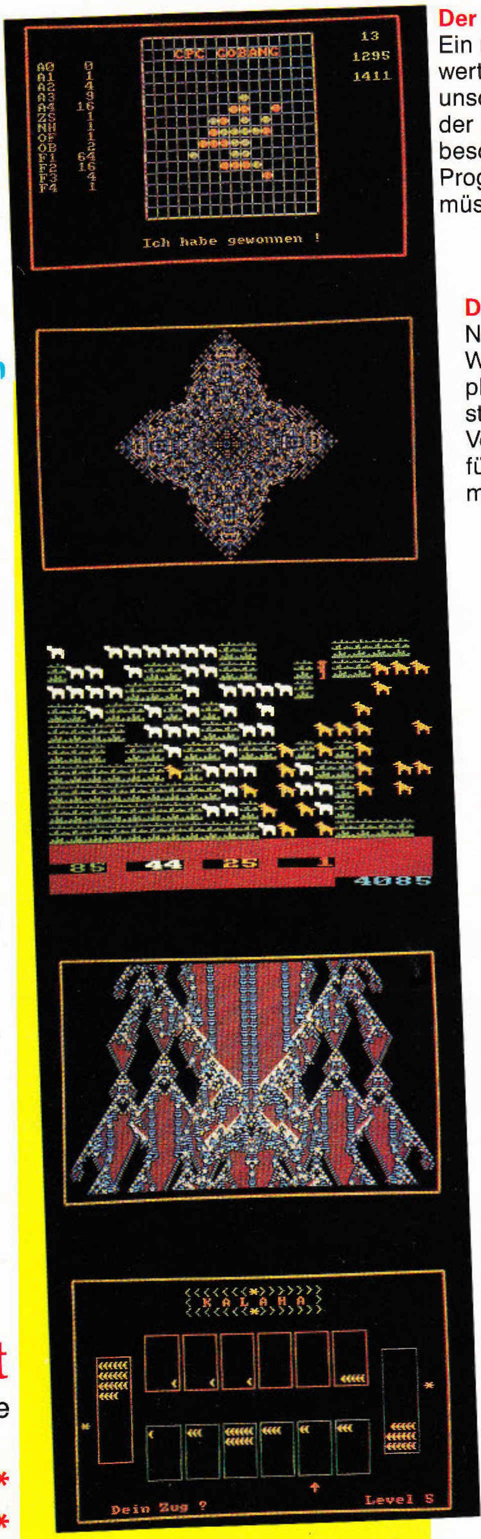
Komplexe Grafik
Sie lernen die mathematischen Grundlagen von Fraktalgrafiken und erarbeiten einen kompletten Fraktalgenerator zur Erzeugung von "Apfelmännchen"-Bildern.

Der Computer lernt
Erleben Sie ein Paradebeispiel für "Künstliche Intelligenz": Das Programm "Minischach" ist lernfähig und wird mit jeder Partie ein bißchen besser.

Wordmaster
Sie raten ein Wort, das sich der Computer ausgedacht hat. Dann rät der Rechner ein Wort, das Sie sich ausgedacht haben. Wer braucht weniger Versuche? Sie werden sich wundern...

Wordketten
Mit einem raffinierten Algorithmus löst der CPC Wortketten-Rätsel. Oder er erfindet neue Rätsel - ganz wie Sie wollen.

Entwicklungshilfe
Eine gut ausgestaltete Simulation versetzt Sie in die Lage eines Entwicklungshelfers, der eine Hungersnot abzuwenden versucht. Aber Ihre Mittel sind begrenzt.



für CPC 464/664/6128

Der Computer als Stratege
Ein raffiniertes System von Spielzugbewertungen macht den CPC zu einem fast unschlagbaren Gegner im Gobang-Spiel, der Ihnen anfangs saftige Niederlagen bescheren wird. Keine Angst - das Programm ist zu schlagen - aber wie, das müssen Sie selbst herausfinden.

Das Spiel des Lebens
Nach natürlichen Prinzipien des Wachstums erzeugt der Computer phantastische, kristallförmige Farbstrukturen. Ändern Sie die Regeln und Voraussetzungen und sehen Sie, was für neue, noch nie dagewesene Formen sich ergeben.

Das Ökologie-Experiment
Der Computer zeigt Ihnen in animierter Farbgrafik einen simulierten Lebensraum mit Weideland, Schafen und Wölfen. Erforschen Sie die ökologischen Gesetzmäßigkeiten, und schaffen Sie ein stabiles, natürliches Gleichgewicht.

Pascal läßt grüßen
Ein erstaunliches Programm enthüllt verborgene Strukturen im sogenannten "Pascalschen Dreieck". Die grafische Darstellung ergibt traumhaft schöne und farbige Musterbilder.

Mit roher Rechengewalt
Mit dem "Brute-Force"-Algorithmus darf der CPC so richtig loslegen und spielt mit roher Rechengewalt in dem afrikanischen Strategiespiel "Kalaha" alle Gegner an die Wand - oder können **Sie** ihn trotzdem schlagen?

Das Software-Experiment

- Siebzehn verschiedene, lauffähige Programme
- 180 Seiten Anleitung

Kassette nur DM 29,– *
Diskette nur DM 39,– *

* Unverbindliche Preisempfehlung. Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4.- DM bzw. für das Ausland 6.- DM Porto und Verpackung.

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Impressum

Herausgeber
Christian Widuch

Chefredakteur
Stefan Ritter

Redaktion
Claus Daschner (cd), Heinrich Stiller (hs), Jürgen Borngießer (jb), Joachim Freiburg (jf)

Redaktions-Assistenz
Anke Kerstan (ke), Susanne Eska (es)

Produktionsleitung
Gerd Köberich

Bereichsleitung
Christian Heckmann (Fotografie/Layout)
Claudia Küllmer (Fotosatz/Lektorat)
Margarete Schenk, Helmut Skoupy (Montage/Reprografie)

Layout
Michael Grebenstein

Fotografie
Heike Reifer, Heike Meister

Fotosatz
Marcus Geppert

Lektorat
Dagmar Wilhelm-Ballhaus

Montage/Reprografie
Dieter Schnobl, Andrea Gundlach

Werbegestaltung
Mohamed Hawa, Petra Küch

Anzeigenverkaufsleitung
Wolfgang Schnell

Anzeigenverkauf für PLZ 1, 4, 5
Gerlinde Rachow (0 56 51) 80 09–53
Wolfgang Brill (0 56 51) 80 09–51

Anzeigenverkauf für PLZ 2 + 3
DMV-Verlagsbüro Hamburg
Ohlsdorfer Straße 34, 2000 Hamburg 60
Sylvia Ehrenpfordt, Tel.: (0 40) 46 12 33
Telefax.: (0 40) 47 43 10

Anzeigenverkauf für PLZ 6 – 8
DMV-Verlagsbüro München
Zaunkönigsweg 2c, 8000 München 82
Telefon: (0 89) 4 39 10 87, Telefax: (0 89) 4 39 10 80
Leitung: Britta Fiebig
Anzeigenverkauf: Monika Schöbel, Jens Dhein, Peter Schätzle

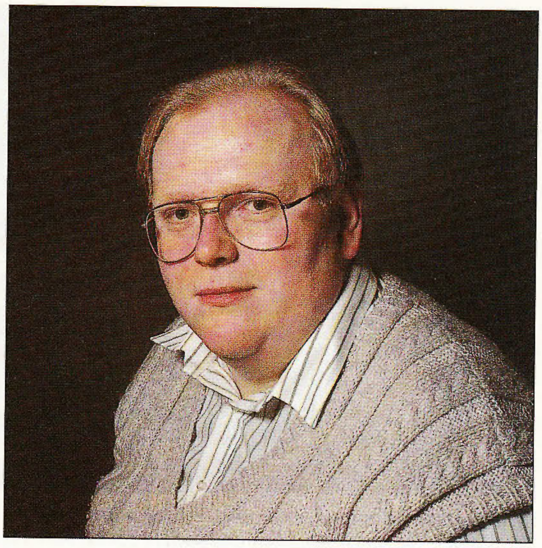
Anzeigenverwaltung und Disposition
Andrea Giese, Karina Ehrlich, Beate Kranz

Anschrift Verlag/Redaktion:
DMV–Daten und Medien Verlag
Widuch GmbH & Co. KG
Fuldaer Straße 6
3440 Eschwege
Telefon: (0 56 51) 80 09 – 0
Telefax: (0 56 51) 80 09 – 33

Vertrieb
Verlagsunion Erich Pabel–Arthur Moewig KG (VPM)
Friedrich-Bergius-Straße 20
6200 Wiesbaden

Druck
Druckerei Jungfer, 3420 Herzberg

Bezugspreise:
Schneider CPC International Sonderheft erscheint unregelmäßig.
Einzelpreis DM 14,–/sfr 14,– /öS 120,–



Software satt!

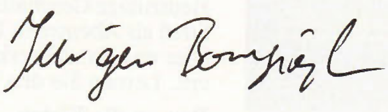
So lautete die Meinung der Redaktion PC AMSTRAD nach Fertigstellung des neuen CPC-Sonderheftes, welches nun vor Ihnen liegt. Endlich vor Ihnen liegt, sollte es wohl besser heißen, denn seit dem letzten Sonderheft ist einige Zeit ins Land gegangen. Zeit, die wir jedoch genutzt haben, um Ihnen wieder eine Fülle an Grundlagen und Programmen zusammenzutragen, die über das normale Maß der PC AMSTRAD International hinausgehen.

Und so finden Sie wieder einiges an 'Futter' für Ihren CPC, unter anderem ein tolles Berechnungsprogramm für Satelliten, deren Bahnen grafisch dargestellt werden, ein Lernprogramm, welches Ihnen die Berechnung von linearen Gleichungen erleichtert, ein anderes Mathematikprogramm, das statt Ergebnissen lieber Apfelmännchen auf den Bildschirm zaubert, sowie Spiele, die Ihnen spannende Unterhaltung am Bildschirm versprechen; ein Umstand, der gerade jetzt die Herbst- und Winterabende verschönern kann.

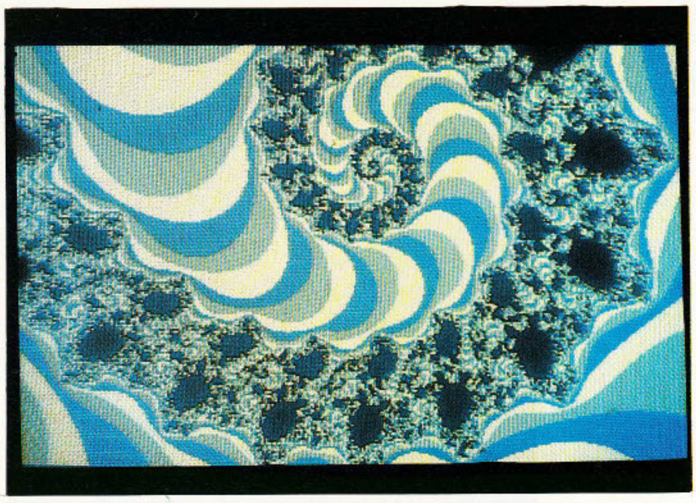
Aber auch Grundlagen über die Datenfernübertragung und CP/M, das 'zweite' Betriebssystem des CPC, sowie eine Menge an Tips und Tricks warten darauf, Ihnen beim Arbeiten mit dem Computer hilfreich zur Seite zu stehen.

Uns bleibt daher nur noch, Ihnen viel Vergnügen bei der Lektüre des Sonderheftes und beim Benutzen der Programme zu wünschen.

Herzlichst Ihr

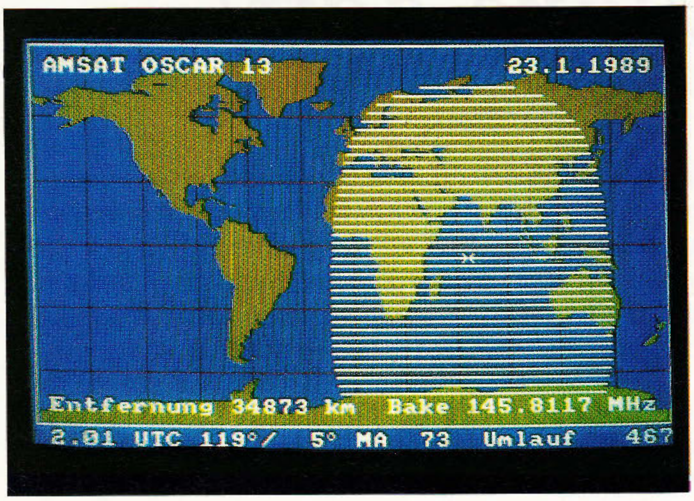


Jürgen Borngießer



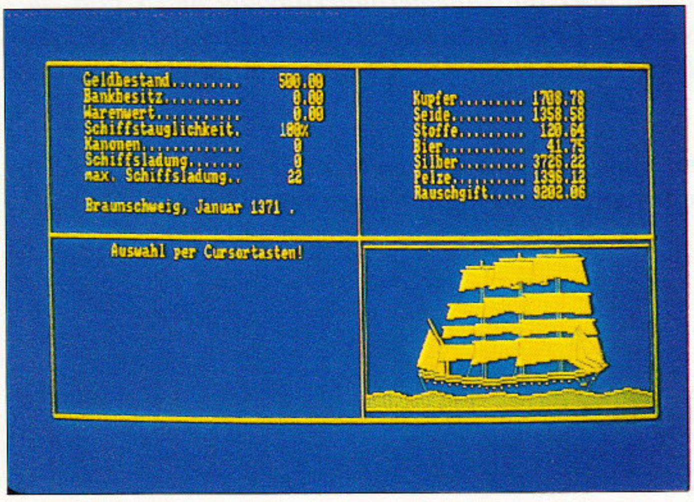
In die wunderbare Welt der Mathematik entführt Sie unser Fraktal-Generator. Finden Sie Apfelmännchen und andere sonderbare Figuren

S. 30



SAT berechnet Ihnen die Umlaufbahn und den Einstrahlbereich eines beliebigen Satelliten und gibt diese Berechnungen grafisch wieder

S. 58



So ein Händler, der hat's schwer – vor allem, wenn er im Hanse-Zeitalter auf Reisen geht. Gehen Sie mit, wenn Sie GOLDDUKATEN und SCHIFFS-KANONEN gerne haben

S. 71

Berichte:

Anwendungen:

Spiele:

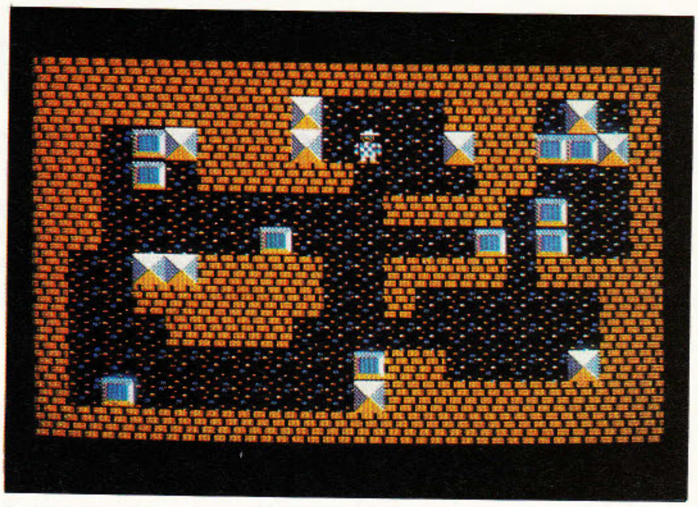
Spiele:

Tips und Tricks:

Hardware:

Grundlagen:

Rubriken:



Verflixte Kiste! Ist schon schlimm, wenn alles im Weg steht, aber wenn man dann noch wenig Platz zum Umräumen hat, wird es gar knifflig. Zumindest bei LETS WORK

S. 85



Wir wissen nicht, was Ihr Hausarzt bei eingerosteten Knochen empfiehlt – wir empfehlen auf jeden Fall die Joystick-Matte im Eigenbau

S. 132



Das Betriebssystem CP/M ist vielen CPC-Benutzern immer noch unheimlich und unbekannt. Daß CP/M jedoch einiges zu bieten hat, zeigt Ihnen unser Grundlagenartikel

S. 142

Nicht nur für Sammler

Softwaresammler, was bieten die?

Immer wieder hört und liest man über Compilationen von älteren Spielen. Mehrere Programme werden zusammen auf einen Datenträger gebracht und zu außerordentlich günstigen Preisen angeboten. Was ist dran an diesen Sammlern? Viel gute Software für wenig Geld oder alte Hüte, neu aufgelegt?

Wissen Sie was eine Ld ist? Nun, die Antwort ist ganz einfach, wenn Sie sich die Ld von der LP her ableiten. Eine LP ist eine Langspielplatte, und somit muß eine Ld (logischerweise) eine Langspieldiskette sein, auf der, ähnlich wie bei einer LP, mehrere verschiedene Titel, in dem Fall Programme, enthalten sind. Im allgemeinen werden diese natürlich(?) nicht Lds genannt, sondern Compilations.

Nun gibt es auch schon ziemlich viele dieser Compilations für die CPCs, und genau hier setzt dieser Bericht an. Er soll eine (kleine) Übersicht über die am Markt befindlichen Compilations bringen. Daß man nicht alle Compilations aufführen oder erwähnen kann, ist klar, da es sich schon um zu viele handelt. Aber die besten unter ihnen haben wir für Sie ausgesucht und ein wenig genauer unter die Lupe genommen.

Daß man bei so vielen "Programmsammlern" auch nicht jedes Programm einem ausführlichen Testbericht unterziehen kann, dürfte ebenfalls verständlich sein. Da die meisten Programme auch schon früher einmal in der PC International getestet wurden, ist dies auch nicht notwendig, da man sich dadurch informieren kann; dennoch haben wir einige der Höhepunkte nochmal ganz kurz erwähnt. Als erstes jedoch muß ich Sie noch auf die nachfolgenden Tabellen hinweisen. Dort stellen wir Ihnen eine Auswahl von Compilations mit ihrem Namen, dem Inhalt und der Preis/Leistungs-Note, die Sie ja auch aus den gewohnten Software Reviews her kennen, vor.

Supreme Challenge

Aber nun geht's los, und zwar gleich mit der, nach meiner Meinung, besten Compilation dieses Berichts – mit Supreme Challenge. Diese Compilation enthält zwar "nur" fünf Programme, von denen aber zumindest vier Stück bereits Evergreens sind. Das beste Programm dieser Ld (Sie erinnern sich?) stellt wohl ELITE dar, die unvergleichliche Weltraum- und Handelssimulation, die vor ein paar Jahren die größten Schlagzeilen machte und auch heute noch ganz oben bei den Spitzenprogrammen steht. Sie müssen als Weltraumhändler so lange Handel mit unzähligen Planeten betreiben, bis sie in die Handelselite aufgenommen werden. Ein Superspiel, das einen immer wieder für Stunden vor den Bildschirm fesselt. Das nächste Programm ist Starglider, ein noch nicht einmal so altes Programm, ebenfalls ein Superhit. Ein voll ausgestyltes Ballerspiel der Extraklasse. Sie fliegen über eine in Vektorgrafik gehaltene Planetenoberfläche und müssen alles abschießen, was Ihnen in den Weg kommt.

The Sentinel und Tetris waren zwar keine so großen Hits wie die beiden vorher genannten Spiele, aber auch sie standen ganz oben in der Software-Hitparade. Bei Tetris handelt es sich um ein Denkspiel, bei dem Sie von oben anfliegende Puzzleteile so ausrichten müssen, daß am Ende der ganze Bildschirm voll ist. In The Sentinel erwarten Sie 10000 Welten, die Sie von Robotern befreien müssen. Dabei geht es nicht ums Ballern, sondern ums Denken, denn Sie müssen es schaffen, über den Robotern zu stehen, um sie zu besiegen. Bei 10000 Welten eine echte Herausforderung. Das letzte Programm dieser Compilation (und nicht nur dieser Compilation) ist ACE II. Ein Kampfflugsimulator, der es aber nicht unbedingt wert ist, auf so einer Spitzen-Ld mit aufgenommen zu werden. Alles in allem ist diese Compilation aber wirklich eine der besten, wenn nicht sogar die beste, die es zur Zeit auf dem Softwaremarkt gibt.

Ausführlicher Test in PC Amstrad International Heft 1/89

Prestige Collection

Auch diese Compilation zählt, wie die vorhergehende, zur Spitzenklasse der angebotenen Compilations. Auf dieser Sammlung sind zwar 'nur' vier Spiele enthalten, aber jedes davon kann mit Fug und Recht als Kleinod bezeichnet werden.

The Eidolon ist ein Spiel, bei dem Sie ein unerforschtes Höhlensystem erkunden müssen. Daß diese Grotten mit allerlei Getier bevölkert sind, braucht wohl nicht extra erwähnt zu werden.

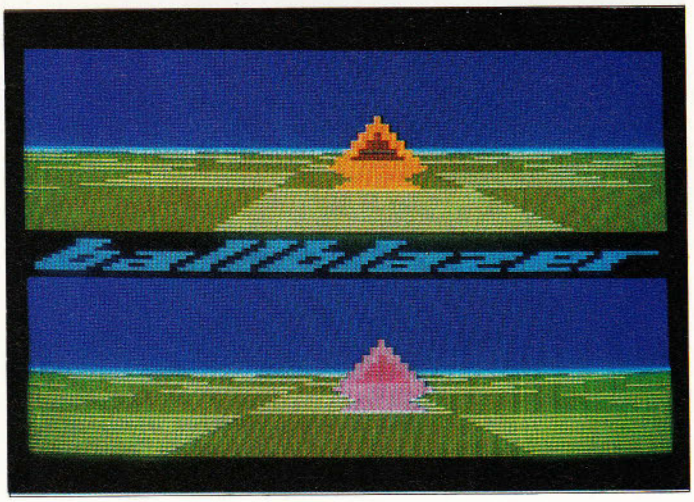
Ballblazer ist das Sportspiel einer fernen Zukunft. Sie müssen versuchen, mittels eines fahrzeugartigen Schlägers einen Energieball in das gegnerische Tor zu befördern.

Rescue on Fractalus. Als Pilot eines Rettungsgleiters müssen Sie havarierte Piloten von der Oberfläche des Planeten Fractalus retten.

Koronis Rift. Als intergalaktischer Schrotthändler haben Sie den Planeten einer lange untergegangenen Kultur entdeckt. Nun gilt es, alle Wertsachen von der Planetenoberfläche zu bergen.

Vier Progamme bester Qualität zum Preis eines einzelnen Spieles, hier heißt es zugreifen, solange der Vorrat reicht.

Ausführlicher Test in PC Amstrad International Heft 9/89

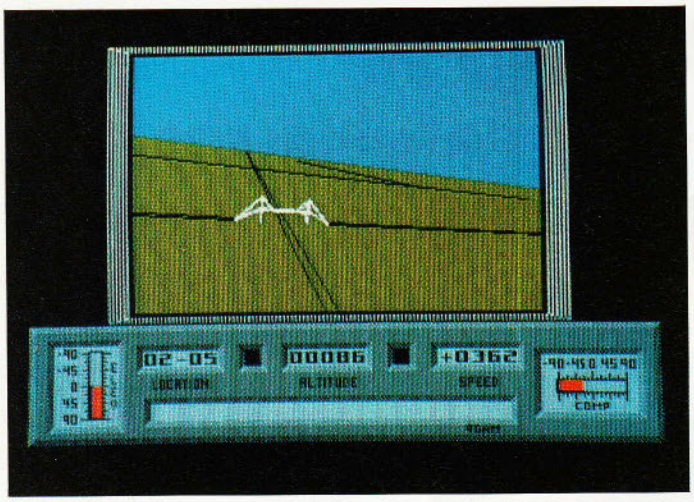


Ballblazer, ein Sportspiel aus ferner Zukunft. Es gilt, einen Energieball in das gegnerische Tor zu befördern. Dieses Spiel finden Sie auf der Prestige Collection

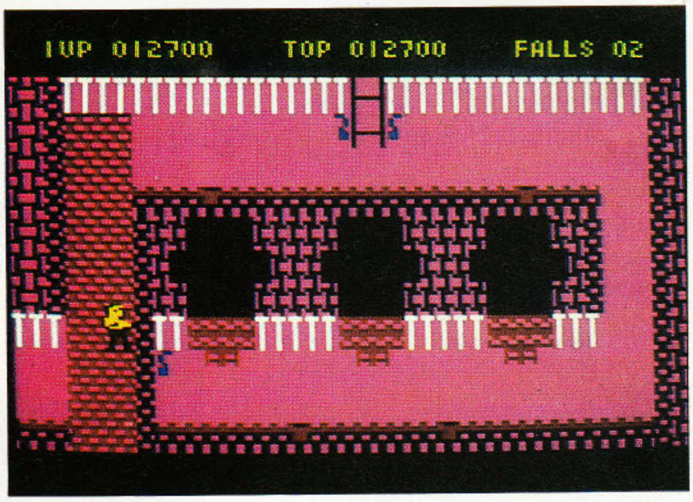
Space Ace

Eine insbesondere für Freunde gepflegter Balleraction empfehlenswerte Sammlung.

Enthaltene Titel wie North Star, Cybernoid und Exolon präsentieren Action und Ballerspaß gehobener Qualität. Grafisch sind alle enthaltenen Titel ein Augenschmaus, die auch in puncto Animation zu dem besten gehören, was es auf dem



Rasante Vectorgrafik und ein gutes Spielkonzept heben Mercenary weit über den Durchschnitt vergleichbarer Spiele. Dieses Spiel finden Sie auf der Command Performance Sammlung



Bruce Lee verbindet Jump and Run mit Elementen aus Karatespielen. Dieses Spiel ist in der Sammlung Karate Ace enthalten

CPC zu sehen gibt. Leider sind einige Titel dieser Compilation auf dem Index.

20 Chart Busters

Eine einmalige Super-Compilation, die zwar zum Teil auch Billigproduktionen enthält, aber alles in allem hält, was sie verspricht.
Hier finden sich Titel wie Ghostbusters, Tau Ceti und The Way of the Exploding Fist, um nur einige der auf dieser Super Compilation enthaltenen Spiele zu nennen. Vom Preis-Leistungs-Verhältnis her ein echter Hit.

The History of Making – The first three Years

Eine etwas andere Ld ist sie schon, denn hier findet man nicht nur elf Programme (+ vier indizierte), sondern man erhält die Geschichte eines Softwarehauses, nämlich die von U.S. Gold. Hier finden sich so bedeutende Hits wie das Golfspiel Leaderboard oder das Labyrinth- und Ballerspiel Gauntlet, bei dem Sie "Tausende" von Feinden abschießen und durch Hunderte von Levels wandern müssen. Auch die Automatenumsetzung Kung Fu Master, in dem Sie sich durch fünf Stockwerke gegen Messerwerfer, Zwerge und Drachen behaupten müssen, um Ihre geliebte Freundin zu befreien; oder die World Games, bei denen Sie gemeinsam mit Ihren Freunden z.B. um die Weltmeisterschaft im Baumstammwerfen, Fässerspringen oder Klippenspringen kämpfen müssen, sind enthalten.

Ausführlicher Test in PC Amstrad International Heft 2/89

Command Performance

Weitere sieben Programme befinden sich auf Command Performance, wobei es sich auch hier wieder um recht gute Programme handelt. Da wäre zunächst der Klassiker Mercenary, bei dem sie auf einem Planeten – mit wunderschöner Vektorgrafik – gestrandet sind und nun versuchen müssen, durch das Auffinden von diversen Ersatzteilen, diesen wieder zu verlassen. Als nächstes ist da noch Xeno, das, wie Ballblazer (siehe oben), ein Sportspiel der Zukunft darstellt. Zwei Spieler versuchen wieder mit Hilfe von zwei schlägerähnlichen Gebilden, einen Ball ins gegnerische Tor zu bringen. Auch dieses Spiel bereitet viel Spaß, der aber nach einer gewissen Zeit schon stark abnimmt. Nicht ganz so gut wie andere hier vorgestellten Lds, aber dennoch empfehlenswert.

Ausführlicher Test in PC Amstrad International Heft 4/89

Star Games II

Eine schon etwas ältere, aber dennoch interessante Ld stellt Star Games II dar. Mit The Eidolon, Ballblazer (siehe oben) und Highway Encounter, einem grafisch und animationstechnisch voll ausgereiften Ballerspiel, bei dem Sie mit fünf Robotern einen riesigen Highway von Aliens befreien müssen, haben sich auch auf dieser Compilation einige gute Spiele versammelt. Trotz ihres Alters kann auch diese Compilation als interessant bezeichnet werden.

Five Star III

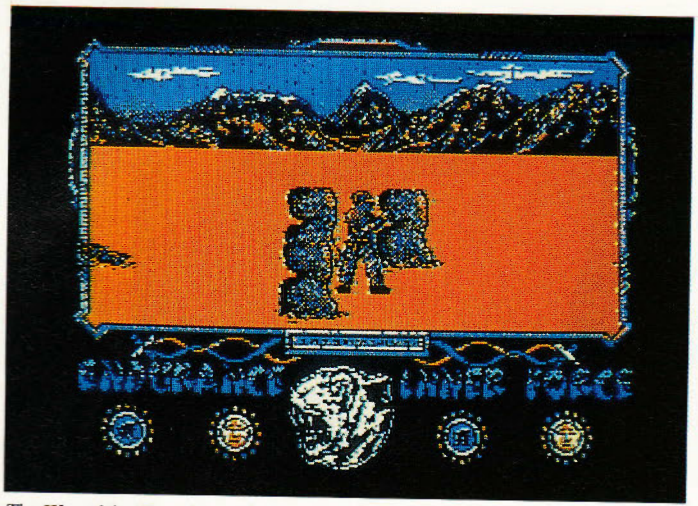
Ebenfalls älteren Semesters ist Five Star III. Eine Compilation, die ihren Namen nicht wegen der Anzahl der enthaltenen Programme trägt, der Name soll Hinweis auf die Qualität der Spiele sein. Und dies nicht zu unrecht.
Die Palette der enthaltenen Programme reicht von Tau Ceti über The Way of the Exploding Fist bis zu Firelord, das sicherlich eines der echten Highlights dieser Compilation ist. In der Rolle des Sir Galaheart müssen Sie ein legendäres Land von einer terrorsüchtigen Herrscherin befreien. In 512 Schauplätzen gilt es, jeder Menge Gefahren zu begegnen.
Ebenfalls enthalten ist Aliens, das Spiel zum Film. Hier gilt es, mit einer Crew aus beinharten Space Marines einen fernen Planeten von üblen Monstern zu befreien.
Hier finden sich ausschließlich Spiele, die einstmals Spitzenplätze in diversen Softwarehitparaden belegten. Auch diese Compilation ist empfehlenswert.

Karate Ace

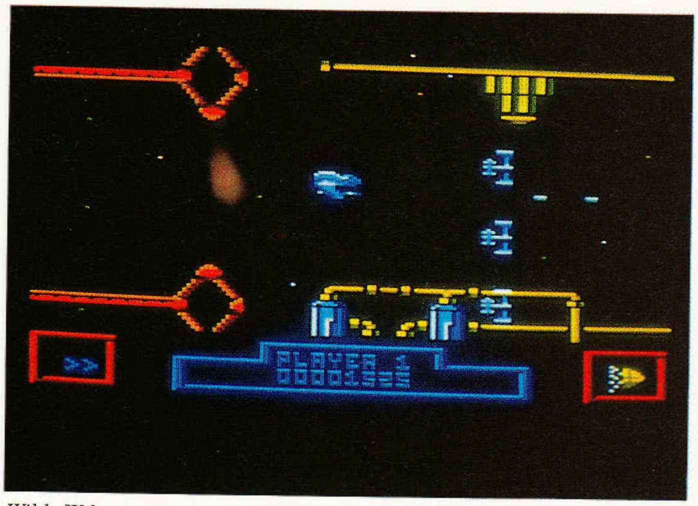
Sieben der besten Karate/Kung-Fu-Spiele auf einem Datenträger. Vom Standardprogramm des Genres 'The Way of the Exploding Fist', das auch auf dieser Sammlung enthalten ist, einmal abgesehen, bekommt der Compilation-Freund alles geboten, was CPC-Software an gepflegter Prügelaction aufbieten kann.
Das beste Programm des Sammlers ist zweifelsohne 'The Way of the Tiger'. Dieses Programm ist seiner Grafik, seiner Animation und seines Scrollings wegen – hier scrollt der CPC vertikal drei unterschiedliche Ebenen – ein Super-Hit geworden.
Auch die Fortsetzung 'Avenger' ist enthalten, obwohl es sich hier weniger um ein Karatespiel, denn um einen Gauntlet-Clon handelt.
Alles in allem eine Sammlung, die viel Spiel fürs Geld bietet.

Game, Set and Match II

Der Nachfolger des Sporthits Game, Set and Match ist noch um Längen besser als der alte selbst, deshalb habe ich ihn auch nicht vorgestellt. Wer kennt nicht Matchday II, den Fußballhit für die Amstrads



The Way of the Tiger ist grafisch, spiel- und programmtechnisch einer der absoluten Höhepunkte der Karatespiele. Dieses Spiel ist in der Sammlung Karate Ace enthalten



Wilde Weltraumaction. Extrawaffen, Levelwächter und Unmassen von Feinden machen aus Zynaps eine echte Herausforderung. Sie finden Zynaps auf der Space Ace Compilation

überhaupt? Da kann man aber wirklich alles verstellen. Man kann die Gegner foulen, den Ball köpfen und die Stärke des Gegners bestimmen. Das einzige, was mir nicht gefiel, war die Geschwindigkeit, die zwar recht schnell ist, aber doch noch ein bißchen schneller hätte gemacht werden müssen. Aber sonst, das Scrolling ist fließend, und die Figuren recht hübsch.

Auch die Motorsport-Fans kommen hier auf ihre Kosten, denn mit Championship Sprint und Super Hang on sind hier gleich zwei gute Motorsportsimulationen zusammen. Bei Championship Sprint, dem Nachfolger von Super Sprint, kann man sogar seine eigenen Rennstrecken zusammenbasteln. Man sieht die Rennstrecke von oben und muß seinen Wagen als erstes ins Ziel bringen. Nicht schlecht!

Auch Super Hang on, ein Motorradrennen, ist nicht von schlechten Eltern, denn es gehört schon einiges dazu, die in schöner Grafik gehaltenen Rennstrecken zu schaffen. Aber hier kommt wirklich kein Sportfreak schlecht weg, denn es sind für so ziemlich jeden genug Sportspiele dabei. Mit zehn Programmen gehört sie zumindest von der Quantität her zu den obersten, aber auch die Qualität ist nicht zu verachten.

Leaderboard Par 3

Was soll man dazu noch sagen? Es sind drei Leaderboardspiele auf einem Pack, und daß Leaderboard zu den besten Golfsimulationen gehört, braucht, glaube ich, nicht extra erwähnt zu werden; und es ist schön, sich mal gemütlich vom Sessel aus durch die Landschaft zu "schlagen". Von der Schlägerauswahl bis zum Berechnen des Windes müssen Sie hier auf alles aufpassen. Golffreunde, holt's Euch!

Flight Ace

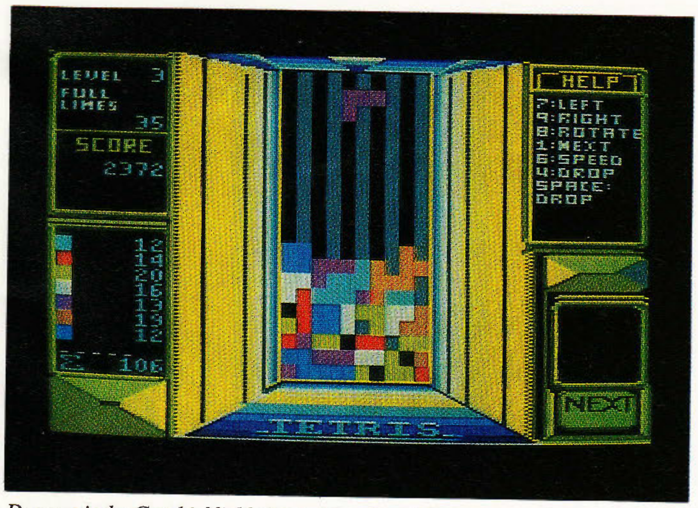
Die letzte hier vorgestellte Ld ist für alle Freunde von Flugsimulatoren gedacht, die auch gerne mal etwas ballern. Hier sind solche Spitzenspiele wie Advanced Tactical Fighter, Tomahawk oder der guten, alten Spitfire 40 zu einer für Flugsimulatorfreunde interessanten Compilation zusammengestellt.

1.) Supreme Challenge Titel: Elite, Starglider, Tetris, The Sentinel und ACE II
Note: 1

2.) Prestige Collection Titel: The Eidolon, Ballblazer, Koronis Rift und Rescue on Fractalus
Note: 1

3.) Space Ace Titel: North Star, Trantor, Cybernoid, Exolon, Zynaps, Venom Strikes Back und Xevious
Note: 1

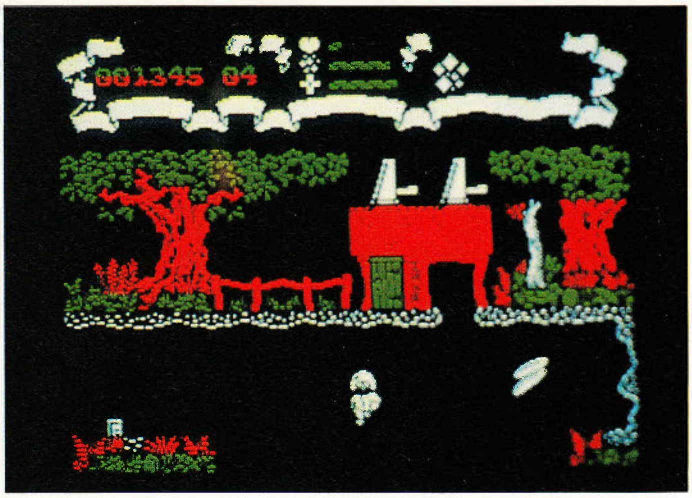
4.) 20 Chart Busters Titel: Ghostbusters, Fa Cup Football, Agent X II, Tau Ceti, Kane, The way of the exploding Fist, Ricochet, Ninja Hamster, Rasputin, Ollie and Lisa, Zolyx, Dan Dare, Formula One Sim., Brian Jack's Superstar Challenge, I-Ball, Park Patrol, Thrust, Harvey Headlanger, War Cars und ein indiziertes Spiel
Note: 1



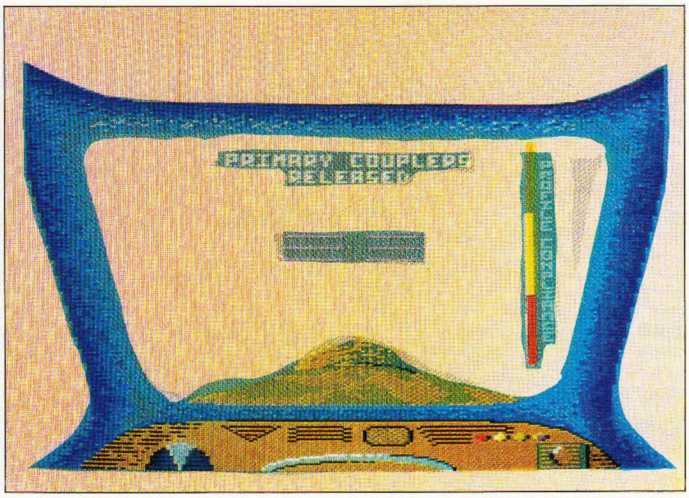
Das russische Geschicklichkeitsspiel begeistert Computerspieler in aller Welt. Ein Highlight, das auf dem Sammler Supreme Challenge enthalten ist



Auf Tau Ceti ist die Hölle los. Nach einer großen Katastrophe gilt es, alle Reaktoren eines fernen, ehemals besiedelten Planeten wieder in Betrieb zu nehmen. Sie finden Tau Ceti auf der Sammlung 20 Chart Busters



Wer Labyrinthspiele mag, liegt bei Firelord genau richtig. Im magischen Mittelalter angesiedelt, gilt es, einer terrorsüchtigen Königin das Handwerk zu legen. Interessiert? Sie finden das Spiel auf der Sammlung Five Star III



Schreckliche Monster aus der Tiefe des Weltraums haben eine Kolonie überfallen. Sie sollen dort nun für Ordnung sorgen. Dieses Spiel finden Sie in der Five-Star-III-Sammlung.

5.) The History of Making Titel: Leader Board, Impossible Mission, Super Cycle, Gauntlet, Infiltrator, Kung Fu Master, Spy Hunter, Road Runner, Bruce Lee, Goonies, World Games und 4 indizierte Programme
Note: 2

6.) Command Performance Titel: Mercenary, Xeno, Hardball, Cholo, Bobsleigh, 10th Frame und Leviathan
Note: 4

7.) Star Games II Titel: Highway Encounter, Knight Games, Ballblazer, Avenger, Trailblazer und The Eidolon
Note: 3

8.) Five Star III Titel: The way of the exploding Fist, Tau Ceti, Firelord, Aliens, Tempest, Trap Door und Strike Force Harrier
Note: 2

9.) Karate Ace Titel: The way of the Tiger, Avenger, The way of the exploding Fist, Kung Fu Master, Samurai Triology, Uchi Mata und Bruce Lee
Note: 2

10.) Game, Set and Match II Titel: Matchday II, Track and Field, Basketmaster, Steve Davies Snooker, Winter Olympiad, Nick Faldo plays the open, Championship Sprint, Super Hang on, Superbowl und Test Match
Note: 2

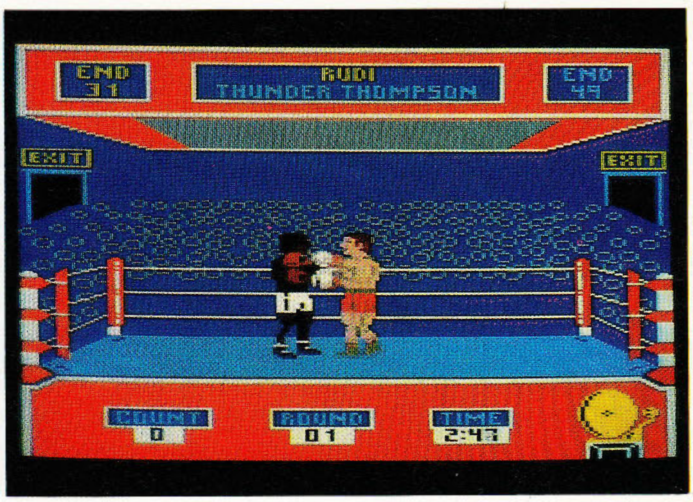
11.) Leaderboard Par 3 Titel: Leaderboard, Leaderboard Tournament, World Class
Note: 2

So, das waren die "Haupt-Lds", aber es gibt noch einige andere, die vielleicht auch nicht so ganz unerwähnt bleiben sollten. Da wäre als erstes mal Gold, Silver, Bronze (s. PC Heft 12/88), mit Summergames I + II und den Wintergames; dann gibt es noch Solid Gold mit den Programmen: Gauntlet, Ace of Aces, Leaderboard, Infiltrator und den Wintergames; die Giants mit California Games, Gauntlet II, Out Run, Rolling Thunder und 720Grad sind auch nicht schlecht; dann noch Game, Set and Match mit Ping Pong, Barry McGuigans World Championship Boxing, Daley Thompson's Supertest, Squash, Konami's Tennis und Snooker, Hypersports, World Series Baseball, Basketball und Super Soccer und Ten Mega Games beinhaltet die "Mega"-Spiele: North Star, Cybernoid, Deflector, Trailblazer, Bloodbrother, Mask II, Tour de Force, Hercules, Masters of the Universe und Blood Valley. Als nächstes käme da noch die Arcarde Force Four, auf der Sie Road Runner, Metrocross, Gauntlet und Indiana Jones wiederfinden, Epics Epyx enthält die Spiele Worldgames, Supercycle, Impossible Mission und Wintergames. Als allerletztes kommen noch die Top Ten, die allerdings nicht so top sind, Saboteur I+II, Critical Mass, Airwolf, Sygma III, Deep Strike, Combat Lynx, Turbo Esprit, Thanatos und Bombjack II. Das war's jetzt aber endgültig. Wir hoffen, daß dieser kleine Bericht Ihnen ein bißchen hilft bei der Auswahl Ihrer nächsten Compilation. Doch schauen Sie erst noch einmal nach, denn es kommen dauernd neue und zum Teil auch noch bessere Compilations auf den Markt.

(Markus A. Litters/hs/cd)



Ein Oldtimer, der nach wie vor für viel Spielspaß sorgt. Wohl eines der ersten Karatespiele überhaupt. The Way of the exploding Fist finden Sie auf der Karate-Ace-Sammlung



Boxweltmeisterschaft mit dem irischen Meister Barry McGuigan. Da kommen Sportspielfans auf ihre Kosten. Dieses Spiel ist in der 'Game, Set and Match' Sammlung enthalten

CPC-SPIELE zu unglaublichen Preisen

CYRUS II Schach

Das bewährte Schachprogramm mit 3D-Display, einstellbarer Spielstärke und deutscher Bedienungsanleitung

Für alle CPCs
Kassette jetzt nur 10,– DM*
Best.-Nr. 130

STARTEST

Action-Adventure für alle CPCs

Kassette jetzt nur 10,– DM*
Best.-Nr. 103
Diskette jetzt nur 19,– DM*
Best.-Nr. 104

Fantastic Four

Vier Superprogramme zum kleinen Preis:

COCKAIGNE – Weltraum-Actionsspiel
TERRANAUT I – Science-Fiction-Adventure
FRUITS – Geschicklichkeitsspiel
TERRANAUT II – Textadventure

Für alle CPCs
Diskette jetzt nur 29,– DM*
Best.-Nr. 1011

CPC Power-Spiele-Paket

18 tolle Spiele für alle CPCs

4 Disketten jetzt nur 49,– DM*
Best.-Nr. 102

SPECIAL OFFERS III

9 Spiele für alle CPCs

3 Disketten jetzt nur 39,– DM*
Best.-Nr. 107

3D-Light Cycle:

Das allseits bekannte und beliebte TRON-Spiel für zwei Personen jetzt wie im Film!! Erleben Sie dieses spannende und abwechslungsreiche Actionspiel jetzt in einer neuen Dimension. Verblüffende und noch nie dagewesene 3D-Effekte sowie professioneller Sound lassen Sie in eine völlig andere CPC-Welt versinken. Diesen Super-Hit muß man 'live' gesehen haben.



3D-Labyrinth:

Das beste Labyrinthspiel jetzt in Super-3D-Qualität. Auf der Suche nach Hinweisen zum Passwort, das den mächtigen Zentralcomputer lahmlegen kann, lauern tausend Gefahren auf Sie. Die räumliche, perfekte 3D-Darstellung, superschneller Grafikaufbau und viele Überraschungen garantieren eine völlig neue Art von Spielvergnügen.

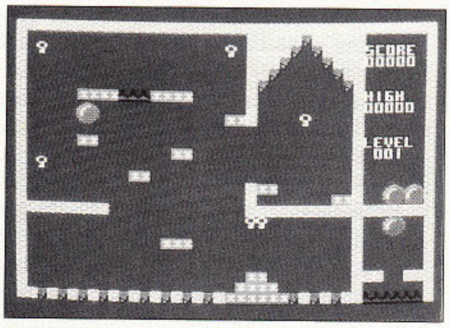
Voraussetzungen:
CPC 464/664/6128 mit Farbmonitor. Das Spiel 3D-Light Cycle kann ohne 3D-Brille auch auf Grünmonitor gespielt werden, Darstellung dann in 2D.

Best.-Nr. 1369
Diskette jetzt nur 39,– DM*

GAME BOX 3

Vier Super-Programme zum kleinen Preis.

Die neue CPC-Spielebox enthält vier ausgesuchte TOP-Programme der Spitzenklasse. Da ist für jeden das Richtige dabei!



1. Alphajet

Lieben Sie Abenteuer und Gefahren im Weltraum? Dann steigen Sie in den neuentwickelten Alphajet und erforschen die interstellaren Sonnensysteme. Vielfältige Gefahren lauern auf Ihrer Mission. Erleben Sie mit Alphajet ein hochklassiges Actionspiel mit außergewöhnlicher Grafik und vielen tollen Effekten.

2. Telefomania

Man schreibt das Jahr 1992. Das sogenannte Datenzeitalter hat begonnen. Alles kann per Datenübertragung von zu Hause aus erledigt werden. Das Problem: Einige Gebiete wurden bei der Vernetzung schlichtweg vergessen. Ihre Aufgabe besteht nun darin, als Einsatzleiter vor Ort diese Vernetzung vorzunehmen. Verhandeln Sie also direkt mit den Hausbesitzern über Tarife und Anschlußmöglichkeiten. Gutes Gespür ist hier gefragt, denn nicht jeder will auch an das Datennetz angeschlossen werden. Mit Telefomania erwartet Sie eine völlig neue Spielidee, die auch Sie begeistern wird.

3. Kampf den Insekten

Übernehmen Sie die Rolle eines berühmten Gärtners. Ihre Aufgabe besteht in der behutsamen Pflege des weltschönsten und mehrfach preisgekrönten Gartens von Lord CPC. Erschwert wird dies durch verschiedenartige Insekten, die immer wieder an Ihren herrlichen Pflanzen nagen und diese zerstören. Durch gezielten Einsatz von Dünger und Sprays (ohne Treibgas) können Sie den unliebsamen Zeitgenossen zu Leibe rücken. Ein erlebnisreiches Geschicklichkeitsspiel für die ganze Familie mit vielen Überraschungen erwartet Sie.

4. Funbouncer

Ihr bester Freund wurde vom bösen Zauberer entführt. Mutig und stark, wie Sie nun einmal sind, machen Sie sich sofort auf die Suche nach Ihrem verschwundenen Kumpan. Als Sie das Schloß des Zauberers erreichen, werden Sie und Ihr Freund in einen feuerroten Ball verwandelt. In dieser Gestalt müssen Sie nun den geheimen Ausgang des Gewölbes finden, um den Fluch zu verlieren. Doch Vorsicht: Vielfältige Gefahren, wie zum Beispiel scharfe Felskanten oder rostige Nägel, machen Ihnen neben anderen Zeitgenossen, die ebenfalls verzaubert wurden, das Leben schwer. Funbouncer ist ein schnelles und farbenfrohes Actionspiel mit vielen Levels und eigenem Bild-Construction-Set.

Für alle CPCs nur als 3-Zoll-Diskette
Best.-Nr.: 1012 **29,– DM***

* Unabhängig von der Anzahl der bestellten Produkte berechnen wir für das Inland DM 4,– bzw. für das Ausland DM 6,– Porto und Verpackung.

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

DFÜ
Ihr Anschluß an die Zukunft
Über die Möglichkeit, Daten in die Ferne zu übertragen

Zählen Sie auch zu den CPC-Benutzern, die neidvoll auf die Besitzer anderer Rechner blicken und mit ansehen müssen, wie die 'mal kurz' eine Mailbox anrufen, um sich die neuesten Tips zum Thema Umweltschutz oder gar Public-Domain-Software herauszuladen? Das muß nicht sein. Haben Sie den Bericht zur Datenübertragung in der PC International 10/88 gelesen? Hier erfahren Sie den Rest der Wahrheit, vom Anfang bis zum ersten 'Login'.

Von Ihrem CPC bis zu einer Mailbox ist es im wahrsten Sinne des Wortes ein weiter Weg. Dieser Weg führt im Normalfall von Ihrem Rechner über die serielle Schnittstelle zum Modem und von dort in das Telefonnetz, um am anderen "Ende" wieder über ein Modem zur seriellen Schnittstelle und von da zum Mailboxrechner zu führen.

Die serielle Schnittstelle – die Verbindung zur Außenwelt

Die serielle Schnittstelle ist dabei unser erster "Wegbegleiter". Das Grundprinzip einer seriellen Schnittstelle ist die Umwandlung von parallelen Daten in serielle und zurück; dies vollzieht in der Regel ein UART (Universal Asynchronous Receiver Transmitter, das heißt Universeller Asynchroner Sender/Empfänger). Das Umwandeln hat den Vorteil, daß, wenn Daten über eine weite Entfernung transportiert werden müssen, bereits drei Leitungen ausreichen, nämlich eine Sendeleitung, eine Empfangsleitung und eine Masseleitung (0 Volt). Würden die Daten hingegen parallel übertragen, wäre eine Vielzahl von Leitungen erforderlich. Die RS232C ist eine solche genormte serielle Schnittstelle, deren Leitungen und Pegel festgelegt sind. Die Pegel betragen gegenüber 0 Volt einmal -12 V für logisch Eins und einmal +12 V für logisch Null. Als Steckverbindung wird für die RS232C ein 25poliger Sub-D-Verbinder, oder neuerdings auch ein 9poliger Sub-D-Verbinder, verwendet. Die DIN unterscheidet 6 verschiedene Arten von Leitungen:

- *Erdleitungen (E1,E2)*
- *Datenleitungen (D1,D2)*
- *Steuerleitungen (S1-S12)*
- *Meldeleitungen (M1-M8)*
- *Taktleitungen (T1-T5)*
- *Leitungen für den Hilfskanal (HD1,HD2,HS2,HM2,HM5,HM6)*

Vermutlich werden Sie sich jetzt fragen, warum das so viele Leitungen sind, wo wir doch vorhin davon gesprochen haben, daß bereits drei ausreichen. Das läßt sich mit den unterschiedlichen Ausbauarten von V.24-Schnittstellen erklären. Der eine Hersteller macht es so, der andere so. Strenggenommen sind das natürlich keine V.24-Schnittstellen mehr, da diese ja genormt sind. Im Grunde wäre es aber auch unsinnig, wenn ein Gerät Leitungen aufweisen würde, die überhaupt nicht benötigt würden; hier beschränkt man sich halt auf das notwendigste und nennt es V.24. Ein 9poliger SUB-D-Verbinder reicht aus, hatten wir gesagt. Somit benötigen wir also nur die Leitungen TxD, RxD, RTS, CTS, DSR, GND, DCD, DTR und RI. Eine ausführliche Beschreibung können Sie der Abbildung 1 entnehmen.

Harte und weiche Hände

Hier wollen wir uns nicht über den Druck Ihrer Hände unterhalten, sondern uns mit der gegenseitigen Steuerung zwischen Endgerät und Modem befassen. Wie bei jeder Kommunikation müssen sich beide Partner so "verstehen", daß zum Bei-

Die wichtigsten Leitungen der V.24 im einzelnen

TxD: Sendedaten (zur DÜE)
Die hier ankommenden Daten werden von der DÜE abgesendet, soweit folgende Leitungen aktiv sind:
DTR – DEE betriebsbereit
DSR – Betriebsbereitschaft (der DÜE)
RTS – Sendeteil einschalten
CTS – Sendebereitschaft (der DÜE)

RxD: Empfangsdaten (von der DÜE)
Die hier ankommenden Daten werden von der DÜE empfangen, soweit folgende Leitungen aktiv sind:
DTR – DEE betriebsbereit
DSR – Betriebsbereitschaft (der DÜE)

RTS: Sendeteil einschalten (zur DÜE)
Mit dieser Steuerleitung wird die DÜE von der DEE informiert, daß diese Daten übertragen will. Ist die Leitung aktiv, schaltet die DÜE ihren Sender ein und ist dann bereit, Daten zu senden.

CTS: Sendebereitschaft (der DÜE)
Über diese Meldeleitung wird die DEE von der DÜE über deren Sendebereitschaft informiert. Es ist also die Quittung auf eine RTS-Anforderung. Ein Aktivzustand bedeutet, daß die DÜE bereit ist, die Daten von der DEE abzusenden.

DSR: Betriebsbereitschaft (der DÜE)
Über diese Meldeleitung wird der DEE mitgeteilt, daß die DÜE betriebsbereit ist und eine Verbindung mit der Gegenstelle besteht.

GND: Signalbezugspunkt
Dies ist die Masseleitung beider Geräte.

DCD: Empfangssignalpegel (der DÜE)
Über diese Meldeleitung erkennt die DEE, daß die DÜE von der Gegenstelle gültige Signale mit ausreichendem Pegel empfängt (das ist ein Datenträgerton), gleichzeitig ist es die Empfängerfreigabe für die DEE.

DTR: DEE betriebsbereit
Mit dieser Steuerleitung steuert die DEE die Anschaltung der Übertragungsleitung (Verbindungsaufbau). Ist diese Leitung inaktiv, wird eine bestehende Verbindung getrennt.

RI: Ankommender Ruf
Über diese Meldeleitung teilt die DÜE dem DEE das Ankommen eines Rufsignals mit (das heißt, ein Verbindungswunsch von der Gegenstelle).
Der Pegel einer inaktiven Leitung ist logisch Eins und entspricht –12V.

Abb.1: Die Leitungsbezeichnungen der seriellen Schnittstelle.

spiel der Rechner dem Modem "sagen" kann: "Halt! Jetzt kann ich nichts empfangen" oder "Achtung! Jetzt möchte ich senden". Genauso muß das Modem auf diese Signale reagieren und darf dann eben z.B. nicht mehr senden. Diese Art der gegenseitigen Steuerung nennt man Handshake, wobei die eine Art, bei der die Steuerung über Leitungen (also hardwaremäßig) erfolgt, Hardware-Handshake heißt. Die zweite Art des Handshakes läuft, wie Sie wahrscheinlich schon vermutet haben, nicht hardwaremäßig, sondern softwaremäßig ab. Hierbei "sagt" der Empfänger dem Sender durch zwei ASCII-Zeichen entweder "Halt" oder "Weitermachen". Das Zeichen X-OFF (CTRL-S, &11) hält die Übertragung an und das Zeichen X-ON (CTRL-Q, &13) läßt sie wieder zu. Diese Art des Handshakes findet auch bei der Bildschirmausgabe vieler Rechner Anwendung. Jetzt wäre nur noch zu klären, wo welche Handshake-Arten verwendet werden. Wie bereits erwähnt, wird das Software-Handshake bereits bei vielen Rechnern angewendet, so hat es sich auch bei den Mailbox-Systemen eingebürgert. Denn verständlicherweise kann zwischen Ihrem CPC und dem Mailbox-Rechner schlecht ein Hardware-Handshake stattfinden, da zwischen beiden Modems nur eine Zwei-Draht-Verbindung (Telefonleitung) besteht. Bei der Verbindungssteuerung zwischen Modem und Rechner verwendet man hingegen das Hardware-Handshake, da hier das Modem direkt an den Rechner angeschlossen ist. Dieses Handshake läßt sich bereits mit zwei der neun wichtigsten Leitungen einer V.24 realisieren. Dummerweise gibt es hierbei zwei Möglichkeiten, die beide nicht das Gelbe vom Ei sind:

1. Die RTS-CTS-Steuerung ist wohl am bekanntesten und regelt die Sendeanforderung der DEE (RTS) und die Sendefreigabe der DÜE (CTS).
2. Die DTR-CTS-Steuerung regelt die Empfangsfreigabe der DEE (DTR) und die Sendefreigabe der DÜE (CTS).

Nur kann es passieren, daß das Modem die Verbindung bei inaktivem DTR-Signal auftrennt. Da hilft nur Ausprobieren, wenn's nicht klappt... Im Normalfall ist die DÜE aber eine reine Übertragungseinrichtung, das heißt, ein empfangenes Zeichen muß durch die DEE sofort abgeholt werden, ehe das nächste empfangen wird, so daß man ruhig das RTS-CTS-Protokoll benutzen kann. Es gibt allerdings auch "intelligente" Modems, die die empfangenen Zeichen speichern können, wenn die DEE nicht empfangsbereit ist; das könnte der DÜE zum Beispiel über RTS mitgeteilt werden.
Bisher haben wir nur über die V.24 und deren Leitungen gesprochen. Aber wie werden denn nun die Bytes übertragen? Auf diese Frage kann man mit einem einzigen Wort antworten: asynchron. Schön, meint der interessierte Leser, sehr schön... wenn man nur wüßte, was es ist!

Alles formatiert

Nun, synchron bedeutet soviel wie gleichzeitig. Hier steht es für getaktet. Also bedeutet asynchron nicht getaktet. Im Klartext heißt das, daß mit der Übertragung der einzelnen Bits zu einem beliebigen Zeitpunkt begonnen werden kann. Jedes Byte wird also bitweise, zuerst das niederwertigste (Bit 0), übertragen. Dem ersten Bit voran wird die Startbedingung (Startbit) gesendet, während nach dem letzten Bit die Stoppbedingung (Stopbit) gesendet wird. Diese beiden Bedingungen sind eindeutig und markieren Anfang und Ende eines Bytes. Zusätzlich kann noch ein Paritätsbit zur Fehlerprüfung mitübertragen werden, dieses muß aber nicht sein. Als Beispiel soll hier Abbildung 3 dienen.

Da ja, wie im Bild zu sehen ist, der Inaktivpegel logisch Eins entspricht, "lauert" der Empfänger des UARTs auf einen logisch Null-Pegel. Daran erkennt er den Anfang eines Bytes und beginnt mit der Umwandlung des ankommenden Datenstroms in parallele Daten. Wie war das doch gleich mit dem Paritätsbit? Dort gibt es die gerade und ungerade Parität. Dabei wird das Paritätsbit immer so "ergänzt", daß die Anzahl der Eins-Bits entweder gerade oder ungerade ist. Auf unser "M" (01001101) bezogen hieße es, daß das Paritätsbit bei gerader Parität nicht gesetzt wäre, weil das "M" in binärer Darstellung 4 Eins-Bits aufweist.

Es ist sicherlich ganz einleuchtend, daß Sender und Empfänger das gleiche Datenformat (so nennt man den Aufbau eines Datums) benutzen müssen, andernfalls gibt's Müll...

Die gebräuchlichsten Datenformate in der DFÜ sind 8 Datenbits, keine Parität und 1 Stopbit oder 7 Datenbits, gerade Parität und 1 Stopbit. In Kurzschreibweise: 8n1 bzw. 7e1. Die verschiedenen Möglichkeiten lassen sich aus folgender Tabelle entnehmen:

7 bit	*no parity*	*1 stopbit*
8 bit	*even parity*	*2 stopbit*
	odd parity	

"Nebensächlichkeiten"

Bislang haben wir die V.24 und die Übertragung von vorne bis hinten durchgearbeitet, es fehlen nur noch die Übertragungsgeschwindigkeit und die Übertragungsart.

Fangen wir mit der Übertragungsgeschwindigkeit an. Sie wird in bit/s (Bit pro Sekunde) angegeben und besagt, wieviel Bit pro Sekunde übertragen werden können. In der DFÜ finden zur Zeit vier verschiedene Geschwindigkeiten Anwendung: 300 bit/s ist die gebräuchlichste und wird von fast allen Mailboxen benutzt.

1200 bit/s hat sich mittlerweile auch schon eingebürgert.

Die Leitungen der V.24 im Überblick

Pin +)	CCITT	DIN	Englische Bezeichnung	Richtung
1	101	E1	protective ground	–
2 (3)	103	D1	transmitted data (TxD)	zur DÜE
3 (2)	104	D2	received data (RxD)	zur DEE
4 (7)	105	S2	request to send (RTS)	zur DÜE
5 (5)	106	M2	clear to send (CTS)	zur DEE
6 (6)	107	M1	data set ready (DSR)	zur DEE
7 (5)	102	E2	signal ground (GND)	–
8 (1)	109	M5	data carrier detect (DCD)	zur DEE
9	–	–	–	–
10	–	–	–	–
11	126	S5	select transmit frequency	zur DÜE
12	122	HM5	2nd data carrier detect	zur DEE
13	121	HM2	2nd clear to send	zur DEE
14	118	HD1	2nd transmitted data	zur DÜE
15	114	T2	transmitter signal element timing	zur DEE
16	119	HD2	2nd received data	zur DEE
17	115	T4	receiver signal element timing	zur DEE
18	–			–
19	120	HS2	2nd request to send	zur DÜE
20 (4)	108	S1	data terminal ready (DTR)	zur DÜE
21	110	M6	data signal quality detector	zur DEE
22 (9)	125	M3	ring indicator (RI)	zur DEE
23	111	S4	data signalling rate selector	zur DÜE
24	113	T1	transmitter signal element timing	zur DÜE
25	–	–	–	–

+) Die Pinnummern in Klammern geben die Pins eines 9- poligen SUB-D- Verbinders wieder.

Abb. 2: Die Datenleitungen der seriellen Schnittstelle und die Pinbelegung

2400 bit/s ist momentan noch Zukunftsmusik, da die Übertragungseinrichtungen dafür noch ziemlich teuer sind.

Die vierte Geschwindigkeit ist eigentlich mehr ein Zwitter; hierbei wird nämlich mit 1200 bit/s empfangen und mit 75 bit/s gesendet. Btx nutzt zum Beispiel diese "billige" Übertragungsgeschwindigkeit, weil diese Modems nicht so aufwendig sind wie die für 1200 bit/s in beide Richtungen.

Und da wären wir auch schon bei der nächsten Nebensächlichkeit, der Übertragungsart. Drei an der Zahl werden unterschieden...

Fangen wir einmal mit dem einfachsten an. Wenn Sie z.B. die Fernbedienung Ihres Fernsehers benutzen und das Gerät daraufhin das gewünschte Programm zeigt, so hat Ihr Druck auf eine Taste dafür gesorgt, daß die Fernbedienung (Sender) mit dem Fernsehgerät (Empfänger) kommuniziert hat; man kann auch sagen, der Sender hat dem Empfänger Daten übermittelt. Ganz einfach. Doch was hat das mit DFÜ zu tun, werden Sie jetzt fragen. Nun, dies nennt man eine Einwegekommunikation, weil die Fernbedienung immer der Sender und das Fernsehgerät immer der Empfänger ist. Es wird bestimmt nicht vorkommen, daß der Fernseher der Fernbedienung etwas mitteilen will... Perfekt fachmännisch nennt man diese Art der Übertragung simplex (oder abgekürzt sx).

Die zweite Art werden Sie vielleicht von einer Türsprechanlage her kennen. Im Ruhezustand ist diese Anlage auf "Hören" gestellt, das heißt, die Übertragung geht nur in eine Richtung. Wollen Sie jetzt "Sprechen", müssen Sie die Sprechtaste drücken, damit der Übertragungsweg umgeschaltet wird. Bei der DFÜ ist das mit dem wechselseitigen "Sprechen" (hier Senden) genauso und wird halbduplex genannt (abgekürzt mit hx oder hdx).

Die dritte Möglichkeit, die noch übrig bleibt, trägt das Kürzel dx wie duplex (oder auch vdx für vollduplex); und da es einem Computer mehr oder weniger egal ist, ob er nun zwei Dinge gleichzeitig tut, gibt es die dritte Kommunikationsart nur bei Maschinen. (Für denjenigen, der sich noch nicht denken kann, was duplex ist: gleichzeitiges Übertragen in beide Richtungen...)

Heutzutage wird in der DFÜ (und von fast allen Mailboxen) ausschließlich die dritte Übertragungsart (duplex) verwendet.

Das wird (an)geboten...

Zuerst das Interface, dieses kommt von Amstrad und beinhaltet neben einem Z80 DART (einem speziellen Wandlerbaustein) noch ein Eprom mit einem Terminalprogramm und diversen RSXes zur Programmierung. Dieses Interface hat allerdings einen Vor- und einen Nachteil: Es läßt sich mühelos vom CP/M-Betriebssystem ansprechen, da die Routinen schon im BIOS vorhanden sind; nur arbeitet es ungern mit der Vortex-RAM-Erweiterung zusammen, da beide das externe ROM #6 benutzen. Dieses Interface ist somit sowohl für den Anwender als auch für den Programmierer gedacht, denn hier gilt: auspacken-einschalten-loslegen. Das Interface kann laut Herstellerangaben an allen CPCs betrieben werden. Abzuraten ist von "Schnittstellenlesungen", die den Joystickport benutzen. Das mag zwar im ersten Augenblick praktisch (und billig) erscheinen. Jedoch ist das nur ein Hilfsmittel und scheitert spätestens bei größeren Übertragungsgeschwindigkeiten. Ein richtiges Hardware-Interface meistert hingegen Übertragungsraten bis zu 19200 bit/s.

Modem und Koppler – zwei ungleiche Brüder

Bisher wurde die V.24-Schnittstelle ausführlich besprochen, jetzt folgt der zweite "Wegbegleiter" – das Modem.

Modem ist ein Kunstwort und setzt sich zusammen aus Modulator und Demodulator. Seine Funktion läßt sich genauso mit zwei Begriffen beschreiben: modulieren von Signalen und demodulieren von Signalen. Modulieren bedeutet, daß der Trägerton durch die zu sendenden Bits seine Frequenz ändert. Diese Modulationsart heißt Frequenzmodulation und wird Ihnen sicherlich vom Radio her bekannt vorkommen. Demodulieren bedeutet dementsprechend, daß der modulierte Trägerton wieder zurück in die einzelnen Bits gewandelt wird. Und genau an dieser Stelle tritt ein kleines Problem auf:



Abb.3: Der Signalverlauf bei der Übertragung, wenn der Buchstabe 'M' übertragen wird

Da das Telefonnetz die eigenen Signale zurückleitet, empfängt der Empfänger die Signale des eigenen Senders. Um diesem abzuhelfen und die DFÜ nicht ganz unmöglich zu machen, werden zwei verschiedene Frequenzgruppen verwendet: Der Anrufer (also Sie) sollten immer im Originate-Mode arbeiten, während der Angerufene (also die Mailbox) immer im Answer-Mode arbeiten sollte. So wird der eigene Empfänger nicht durch den eigenen Sender gestört.

Der Grund, weshalb überhaupt ein Modem benötigt wird, ist relativ einfach: Jeder Rechner, also auch der CPC, arbeitet mit digitalen Signalen, kennt somit nur zwei logische Zustände. Das Telefonnetz (dazu kommen wir später) arbeitet hingegen analog, denn hier wird ja in erster Linie Sprache (Töne) übertragen. Das Modem dient also als Wandler zwischen zwei "Welten", auf der einen Seite digital und auf der anderen analog. Die Geschwindigkeit, mit der Modems arbeiten können, hängt natürlich von der Bandbreite auf der Analogseite ab. Das Telefonnetz hat eine Bandbreite von 3,1 kHz und begrenzt die Übertragungsgeschwindigkeit daher auf 2400 bit/s. Zwar gibt es auch Modems, die mit höheren Geschwindigkeiten arbeiten, jedoch wird da nicht mehr mit der Frequenzmodulation gearbeitet. Eine weitergehende Erklärung würde hier allerdings zu weit führen. Worin besteht aber eigentlich der Unterschied zwischen einem Modem und einem Akustikkoppler? Diese Frage läßt sich recht einfach beantworten. Werfen wir dazu einen Blick in die Definition der Deutschen Bundespost:

Dort wird zwischen "galvanisch" (das bedeutet leitend) verbundenen und akustisch gekoppelten Modems unterschieden. Demzufolge sind also Akustikkoppler auch Modems? Im Grunde genommen ja, denn sie erfüllen ebenso das oben beschriebene Prinzip wie Modems. Den "kleinen Unterschied" gibt es allerdings dennoch: Da bei einem Akustikkoppler die Übertragung akustisch gekoppelt abläuft (der Telefonhörer wird dazu in den Koppler gelegt), können natürlich bei lauten Umgebungsgeräuschen Übertragungsfehler auftreten. Ein Modem ist hingegen direkt an die Telefonleitung angeschlossen (einfach die beiden Leitungen des Modems an die Telefonleitung anklemmen), so daß hier ruhig ein Preßlufthammer bei der Arbeit sein kann; solange er die Leitung nicht trifft, gibt's auch keine Probleme...

Ein Modem ist auch komfortabler zu bedienen als ein Akustikkoppler: Bei diesem muß die Verbindung per Hand aufgebaut werden (also gewählt), während

VIEL INFORMATION für so wenig Geld!

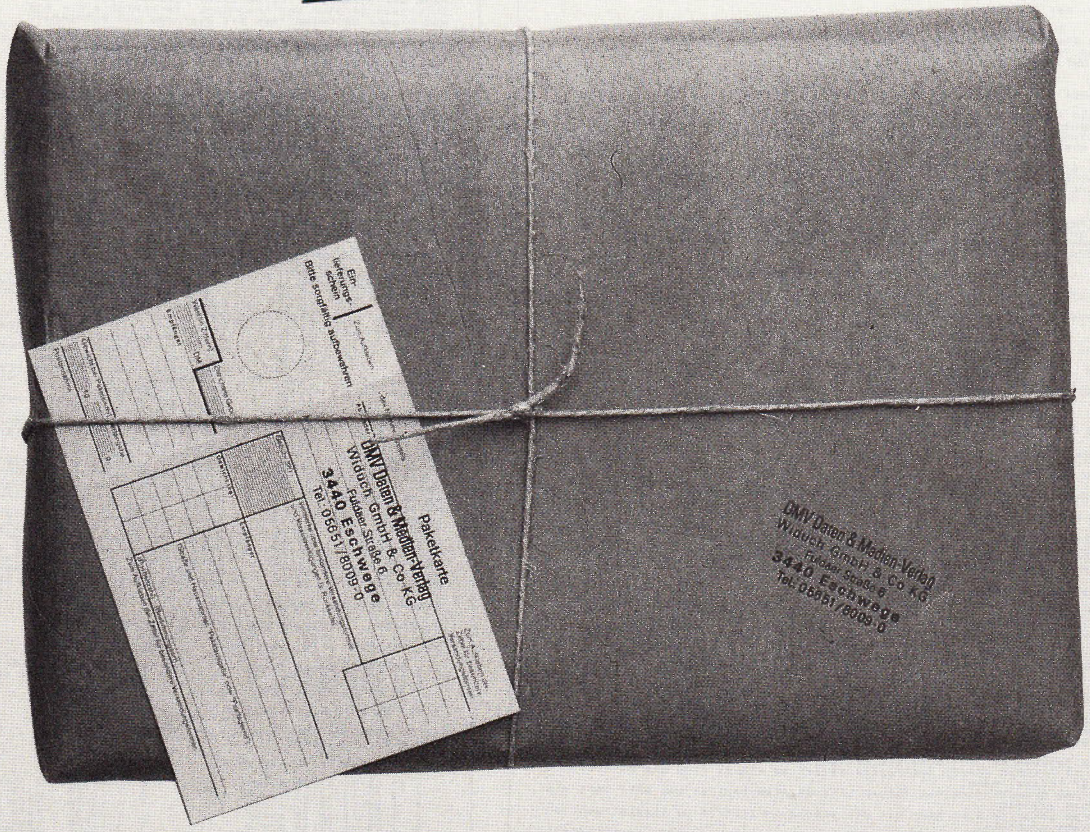
NUR 50,– DM

kostet Sie das Ultra Pack

Jetzt bestellen:

Begrenzte Restmenge

nur solange der Vorrat reicht



alle Ausgaben von PC Amstrad/Schneider Int. des Jahrgangs 1987

+ 3 Ausgaben von PC International des Jahres 1986 aus unseren Restbeständen

+ 2 Sammelordner zum Archivieren

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag – Postfach 250 – 3440 Eschwege

DMV Daten- und Medienverlag

ein Modem dies auf Befehl tut. Fazit: Mit einem Modem geht's einfacher und macht auch mehr Spaß.
Ein nicht ganz unwesentlicher Punkt ist, wie immer, das liebe Geld. So werden Akustikkoppler für etwa 300,– DM angeboten, während Modems ein Vielfaches davon kosten. Der einzige Ausweg aus diesem Dilemma wäre ein Hayes-kompatibles Modem, welches bereits für den Preis eines Akustikkopplers zu haben ist. Der Vorteil liegt klar auf der Hand: Modemkomfort zum Preis eines Akustikkopplers. Dabei haben Hayes-kompatible Modems noch den weiteren Vorteil, daß sie sich durch ihre Verbreitung zu einem Quasistandard entwickelt haben und so von vielen Software-Paketen unterstützt werden.
In diesem Zusammenhang muß ausdrücklich darauf hingewiesen werden, daß das Betreiben von Modems (oder Kopplern) ohne Genehmigung des Zentralamtes für Zulassungen im Fernmeldewesen an öffentlichen Fernmeldenetzen in der Bundesrepublik Deutschland und in Berlin (West) verboten ist und strafrechtliche Folgen nach sich ziehen kann. Das war schönstes Postdeutsch und mußte auch 'mal sein.

Netze...

Sind Sie bis hierher gefolgt? Dann ist es jetzt genug der grauen Theorie, gehen wir gleich zur Praxis. Doch vorher sollten Sie noch ein paar allgemeine Dinge kennenlernen. Da dürfte zum Beispiel die Frage aufgetaucht sein, wie ich mich in eine Mailbox "einlogge" (von 'Login', das ist die Zugangsprozedur, die DBP spricht hier von anschalten). Wie bekomme ich Nummern von Mailboxen? Und, und, und... All diese Fragen wollen wir hier Schritt für Schritt klären. Zuerst der Begriff Netz. Natürlich ist damit kein Fischernetz gemeint, sondern eine Verbindungsmöglichkeit zwischen jeweils zwei Netzendpunkten. Netze (auch das Telefonnetz) sind im allgemeinen in sogenannte Netzknoten (so werden Vermittlungsstellen genannt) aufgeteilt. Um beim Telefonnetz zu bleiben, gehen wir einmal davon aus, Sie wählen die Telefonnummer 02203. Die erste Null bedeutet, daß Sie die Fernvermittlungsstelle ansprechen. Die erste Zwei baut die Verbindung zur Zentralvermittlungsstelle Düsseldorf auf. Dort sorgt die zweite Zwei dafür, daß die Verbindung zur Hauptvermittlungsstelle Köln geht. Die zweite Null spricht nun die Knotenvermittlungsstelle Köln an und die Drei gibt die Ver-

```
Bitte 3 Punkte eingeben ...

Bitte Namen eingeben: gast

Guten Tag, Mailbox-Gast!

Der W D R - Computerclub begr}~t Sie
mit einem Siemens PC-MX2 Mehrplatzsystem
am Sonntag, den 13-11-88 um 12:11:52
Sie sind der 2359. Anrufer auf Port tty12.


>>> Achtung! 483 Login-Versuche mit falschem Passwort!

Als Gast in KomCom haben Sie uneingeschraenkten Zugriff auf alle
Informationen. Sie koennen an jeden Teilnehmer Post verschicken.
Aus rechtlichen Gruenden koennen Nachrichten von Gaesten aber
erst von uns freigegeben werden.

Sie haben allerdings die Moeglichkeit mit dem Befehl ANTRAG einen
Teilnehmer-Antrag zu stellen. Eingetragene Teilnehmer bekommen
ein eigenes Postfach (durch Kennwort geschuetzt) und koennen
ohne Zeitverzug Nachrichten innherhalb des Systems verschicken.

Nach Antragstellung bekommen Sie innerhalb weniger Tage vom WDR
mit der "gelben" Post eine Bestaetigung des Antrages mit dem
ersten Kennwort. Es koennen nur Antraege bearbeitet werden, welche
alle geforderten Angaben enthalten.

- Dieser Text laesst sich durch Steuerzeichen abbrechen -

Befehlsliste mit '?'. Hilfe durch 'HILFE' !

In folgenden "Brettern" nachschauen:

amiga           (11-11-88)      angebote        (10-11-88)
anregungen      (11-11-88)      apple           (09-11-88)
atari           (03-10-88)      btx             (04-11-88)
commodore       (16-10-88)      df}             (10-10-88)
gesuche         (09-11-88)      hardware        (21-10-88)
mailboxwerbung  (11-11-88)      msdos           (30-10-88)
pascal          (05-10-88)      schneider-cpc   (25-10-88)
sinclair        (05-11-88)      system-info     (28-10-88)
termine         (11-11-88)      videodat        (09-11-88)
videotext       (10-10-88)

Keine neuen Nachrichten.

Befehl: b schneider

Brett schneider-cpc: Brett f}r Schneider CPC/Joice-Benutzer

Nr.  * Datum Zeit  Abs./Empf.        Kb  Betreff

  1 27 05-10 13:31 sysop              2  Ascii-Bild-Konverter
  2 18 05-10 13:32 sysop              6  Galgenbaum-Raten
  3 25 20-10 20:31 gast               1  handshake-probleme
  4 17 25-10 08:25 gast               1  Nochmal Aufruf

(schneider-cpc) Befehl: 1 3 4

Nachricht Nr.:    3
Nachricht von:    gast
Betrifft:         handshake-probleme
Abgesandt am:     20-10-88, 20:31:55
Empf{nger:        schneider-cpc
Abfrage Nr.:      26
Text:

Hallo liebe User des CPC 464/664 oder 6128.
Ich habe mit meinem CPC 664 ein Problem von dem ich gerne wuesste
ob es auch anderweitig bekannt ist und ob jemand abhilfe kennt..
die sache ist die dass nach einem return einer mailbox bei 1200 baud
uebertragungsrate auf meinem bildschirm immer das 2.zeichen der neuen zeile
fehlt.
dies kann behoben werden indem die mailbox nach einem carrige-return
ein paar (2 reichen schon) null-bytes sendet so wie es auch die mailbox
vom lothar (ctbbs) macht .
denn der cpc braucht nach nem return btreibsystem-bedingt was mehr zeit als
fuer andere zeichen.
mit anderen boxen so auch hier mit der komkom im 1200/75 bd zugang habe ich
wieder diese probleme.
die schnittstelle die ich verwende ist uebrigens die schneider-schnittstelle
das problem is ausserdem nicht abhaengig von der software..bei telecom 1000
macht mein compi die selben facksen...
falls jemand das selbe probelm hat und abhilfe kennt bitte nachricht hier oder
besser in die ctbbs vom lothar  ..tel.: Koeln/ 8903821
viel gruesse de juergen aus koeln-braunsfeld
,ende

Nachricht Nr.:    4
Nachricht von:    gast
Betrifft:         Nochmal Aufruf
Abgesandt am:     25-10-88, 08:25:41
Empf{nger:        schneider-cpc
Abfrage Nr.:      18
Text:

                                          Koeln, den 24.10.88

Abs.:
Juergen Bartz
zu erreichen in der:
CTBBS Tel.: 0221-8903821
      300/1200 8N1


Hallo Schneider CPC-User,

ich mache nun seit ein paar Wochen Dfue und bin der Meinung,
dass die Moeglichkeiten die dieses Medium in sich birgt bei
weitem nicht ausgenutzt werden.

Speziell als Schneider CPC-User ist man wegen der geringen
Praesenz auf eine Zusammenarbeit angewiesen.

Ich moechte nun Vorschlagen, dass sich die CPC-User im
Raum Koeln auf eine fuer diesen Zweck gemeinsahme
Mailbox einigen.
```

```
Fuer diesen Zweck rufe ich dazu auf bei Interesse die
CTBBS anzuwaehlen und mir folgende Informationen zu
hinterlassen:

Name des Absenders, verwendeter Computertyp,
Interessenbereich der Anwendung, Vorschlag einer Box,
und eigene Heimatmailbox mit Username und Rufnummer.

Ich schlage in diesem Zusammenhang die CTBBS als Box vor,
da diese noch recht unausgelastet ist, somit nicht mit
langen Wartezeiten zu rechnen ist, und diese auch auf
einem CPC betreiben wird.

Ich werde nach einigen Wochen (insofern die Resonanz auf
meinen Aufruf aktzeptabel war) eine Statistik in die
CTBBS schreiben aus der sich dann der weitere Werdegang
der Aktion ableiten laesst.

In der Hoffnung reger Resonanz verbleibe ich
mit freundlichen Gruessen
Juergen Bartz
CTBBS: 0221/8903821
       300/1200 8N1


(schneider-cpc) Befehl: b *

Liste der "Schwarzen Bretter":

68000-assembler 01-01      Forum f}r 68k-Maschinenprogrammierung
80x86-assembler 01-01      Forum f}r 80x86-Maschinensprache
adventure       01-01      Infos und L|sungen zu Adventure-Spielen
amiga           11-11      Brett f}r AMIGA-Benutzer
angebote        10-11      Verk{ufe/Tauschangebote (nicht gewerblich!)
anregungen      11-11      Ideen und Kritik zur Mailbox
apple           09-11      Alles um Apple-II Computer
assembler       01-01      Forum f}r Assemblerprogrammierung
atari           03-10      Alles f}r Atari Computer
basic           01-01      Forum f}r Basic-Programmierer
basicode        01-01      Basicode-II Libraries und Anwendung
bilder          01-01      WDR-Bild}bertragung
btx             04-11      Infos aus und um Bildschirmtext
c-freaks        01-01      Forum f}r C-Programmierer
commodore       16-10      Alles f}r Commodore 64/128/...
computerclub    01-01    # Neuigkeiten vom WDR-Computerclub
df}             10-10      Neues aus der DF]-Szene
gesuche         09-11      Kauf- und Tauschgesuche
hardware        21-10      Hardware -Infos und -Tricks
kaleidoskop     01-01    # Neuigkeiten von KALEIDOSKOP
mac             01-01      Alles um den Apple-Macintosh
mailboxwerbung  11-11      Werbeforum f}r ancere Mailboxen
msdos           30-10      Brett f}r MS-Dos-Anwender
pascal          05-10      Forum f}r Pascal-Programmierer
schneider-cpc   25-10      Brett f}r Schneider CPC/Joica-Benutzer
sinclair        05-11      Forum f}r Sinclair-Benutzer
system-info     28-10    # Informationen und Hinweise des Systembetreibers
termine         11-11      Terminedaten zu Messen, Clubtreffs, etc ..
videodat        09-11      Neuigkeiten um das Videodat-System
videotext       10-10      Informationen aus und zum Videotext

(schneider-cpc) Befehl: b btx

Brett btx: Infos aus und um Bildschirmtext

Nr.  * Datum Zeit  Abs./Empf.        Kb  Betreff

  1 89 17-10 10:22 sysop             18  Verzeichnis der \-BTX Terminals
  2 60 17-10 10:23 sysop              1  BTX-Statistik vom 16.10.1988
  3 18 04-11 10:23 sysop              5  Bedienungsanl  f}r franz. Teletel-Syste

(btx) Befehl: b videodat

Brett videodat: Neuigkeiten um das Videodat-System
Betreuer: m.wiegand

Nr.  * Datum Zeit  Abs./Empf.        Kb  Betreff

  4 17 09-11 12:12 sysop              2  Kaleidoskop 11/88 - Inhalt & Sendetermi
  6  6 09-11 12:16 sysop              2  K.11/88 - Quiz
  7  8 09-11 12:17 sysop              5  K.11/88 - Info Herzschrittmacher
  8  6 09-11 12:18 sysop              6  K.11/88 - Info zum Flimmern der Sterne
  9  7 09-11 12:19 sysop              5  K.11/88 - News
 11 15 09-11 12:22 sysop             12  K.11/88 - GWBASIC Programme
 12  7 09-11 12:23 sysop              6  K.11/88 - C-64 Basic "Vier gewinnt"
 13  4 09-11 12:24 sysop              7  K.11/88 - Amiga C "Dir Lister & Sorter"
 14  2 09-11 12:25 sysop              3  K.11/88 - Tip K}hlschrankentsorgung
 15  4 09-11 12:26 sysop              9  K.11/88 - Filmtext "Grillen"

(videodat) Befehl: b videotext

Brett videotext: Informationen aus und zum Videotext

Nr.  * Datum Zeit  Abs./Empf.        Kb  Betreff

  1 66 10-10 13:12 sysop             12  Erhebungsergebnisse des Politbarometers
  3 46 10-10 13:15 sysop              1  Verbrauchertips zu Recht und Ern{hrung
  4 53 10-10 13:19 sysop              1  Wechselkurse vom 9.10.88

(videotext) Befehl: ?

Folgende Befehle stehen zur Verf}gung:

?             Antrag        Brett         Ende
EInstellen    Hilfe         Inhalt        Kommentar
Lesen         Mailbox       Senden        Teilnehmer
Zeit

(videotext) Befehl: e

Verbindung beenden (Ja/Nein)? j
Anruf um 12:21:31 Uhr beendet !
Dauer der Verbindung: 9 Minuten, 39 Sekunden

Auf Wiedersehen, Mailbox-Gast !
```

Abb.4: Das Mitschnitt-Protokoll der KomCom-Mailbox

bindung zur Ortsvermittlungsstelle Köln-Porz weiter. Das waren jetzt eine Menge neuer Begriffe, aber das Netzprinzip ist denkbar einfach: Durch die Aufteilung in mehrere Ebenen von Vermittlungsstellen (Netzknoten) entsteht eine Art Leitweg, der auch von den Postleitzahlen her bekannt ist. Selbstverständlich gibt es noch mehrere Netze, z.B. Telex, Teletex, Datex...

Das Datex-P-Netz hat in letzter Zeit immer mehr an Bedeutung gewonnen, da über dieses Netz viele Rechenzentren zu erreichen sind, private Mailboxen hingegen "hängen" eher am Telefonnetz... Das waren jetzt alles Netze, in denen jeweils zwei Endgeräte direkt miteinander verbunden sind. Doch es gibt noch eine zweite Art von Netzen. Diese zweite Art kann man mit dem Briefpostnetz vergleichen. Es dient genauso dem Nachrichtenaustausch zwischen zwei Teilnehmern, ohne daß diese direkt miteinander verbunden sind. Die Bezeichnung lautet, nebenbei bemerkt, "Message Handling Systems", und es gibt eine Unzahl davon; das bekannteste ist das UUCP-Netz, an das auch Universitätsrechner angeschlossen sind...

Wir bleiben bei den "direkten" Netzen und da besonders beim Telefonnetz. Obwohl es nicht für die Datenkommunikation gedacht ist, weil es ja ein Analognetz ist, hat es sich in kürzester Zeit zum beliebtesten Medium für die DFÜ entwickelt. Die Hauptgründe dafür sind wohl der weltweite Zugang, die vergleichsweise niedrigen Gebühren und die Verbreitung des Telefons.

```
PRO1: Verbindung hergestellt mit 0221 249508

ProNet Mailbox (PRO1)

Name? GAST


Guten Morgen, Mailbox-Gast!

*** Anhalten: Ctrl-S  UEberspringen: Ctrl-C  Abbrechen: Ctrl-X ***

Willkommen in der ProNet Mailbox !

Sie befinden sich als Gast in unserem Mailbox System.  Es  stehen
Ihnen somit nur wenige der  Mailbox-Befehle zur  Verfügung.  Mit
diesen koennen Sie zwar das System  durchlaufen und sich ein Bild
von Inhalt und Aufbau der ProNet Mailbox machen,  Sie koennen je-
doch  keine  Nachrichten  lesen bzw.  an schwarze Bretter oder an
Teilnehmer der Mailbox versenden!  Durch Eingabe eines ? erhalten
Sie eine Liste der für Gaeste gueltigen Befehle. Mit HILFE <Bef-
ehl> erhalten Sie Erlaeuterungen zu diesen Kommandos.

Die  Mailbox  selbst ist aehnlich wie die GeoNet Mailbox (ehemals
IMCA) aufgebaut.  Fuer alle,  die GeoNet nicht kennen,  hier eine
kurze Beschreibung einiger Mailbox-Merkmale:

 o Befehlsorientierte,  "GeoNet-kompatible"  Benutzeroberflaeche.
 o Alle Nachrichten stehen in  sogenannten  "schwarzen  Brettern"
   und sind einzeln abrufbar. Auf Wunsch koennen  solche  Bretter
   geschlossenen Benutzergruppen zur Verfuegung gestellt werden.
 o Anzeige  aller  Nachrichten, die nach Ihrem letzten Mailboxbe-
   such  neu eingetroffen sind.
 o Versand von  persoenlichen  Nachrichten  mit Empfangsbestaeti-
   gung: Sie bekommen angezeigt, ob und  wann Ihre  Nachricht vom
   Empfaenger gelesen wurde.
 o Ruecknahme von  veralteten oder ueberfluessig gewordenen Nach-
   richten.
 o Weiterleiten  von Nachrichten an andere Benutzer oder Bretter.
 o Moegliche Bestimmung eines 'Vertreters', an  den  persoenliche
   Post bei Abwesenheit weitergeleitet wird.
 o Nachrichten-Verteiler fuer den Postversand innerhalb geschlos-
   sener Benutzergruppen.
 o Moegliche  Definition eines  persoenlichen  Ansage-Textes, der
   jedem angezeigt wird, der Ihnen eine Nachricht zukommen lassen
   moechte.
 o Umlautemulator/-Wandler,  mit dessen Hilfe Sie Umlaute, sofern
   Ihr Terminal nicht dafuer ausgelegt ist,  konvertieren  lassen
   koennen.
 o XModem/Kermit File-System mit viel Public-Domain Software fuer
   MSDOS, Atari ST, C64/C128, CBM610, AMIGA, CP/M, VAX/VMS, UNIX,
   IBM/VM, u.a.
 o Umfangreiche Hilfstexte zu allen Befehlen.
 o Telesatz-Dienst.  Es besteht die  Moeglichkeit,  Druckvorlagen
   via Mailbox erstellen zu lassen.
 o Mailboxbetrieb auf 300 oder 1200 Baud Voll-Duplex.
 o In  Vorbereitung:  Anschluss an  das  weltweite  Unix-Netzwerk
   'Usenet'.

Um  die  ProNet Mailbox nutzen zu  koennen muss man eingetragener
Teilnehmer sein.  Noch koennen Sie kostenlos Benutzer des Systems
werden.  Zugunsten einer  schnellen und einfachen Bearbeitung der
Benutzerantraege koennen Sie sofort "online" in der Mailbox einen
Teilnehmerantrag  abschicken.  Dazu  werden die folgenden Angaben
benoetigt:

 - Gewuenschter Benutzername (max. 15 Zeichen, keine Leerzeichen)
 - Gewuenschtes Passwort (4-15 stellig, keine Leerzeichen)
 - Vorname, Name
 - Strasse und Hausnummer
 - Wohnort (mit PLZ)
 - Telefonnummer
 - Computertyp

Nach Moeglichkeit  sollte der Benutzername  aus dem  abgekuerzten
Vornamen  und Ihrem Nachnamen gebildet werden,  ohne  Leerstellen
dazwischen (Beispiel: aus Arthur Dent wird A.DENT).  Sie  koennen
stattdessen auch  ein Pseudonym waehlen,  sofern  Ihnen dies eher
zusagt.  Erlaubte Sonderzeichen sind: Bindestrich "-", Underscore
"_" und Asterisk "*". Geben Sie bei Pseudonymen nach Moeglichkeit
eTnen Ersatzwunsch an - viele "Namen" sind bereits vergeben. Ihre
persoenlichen  Daten  werden  von uns  gemaess den entsprechenden
Datenschutz-Bestimmungen behandelt und unterliegen der Geheimhal-
tung.

Sie sollten  sich innerhalb von 4 Wochen nach Ihrem Antrag  unter
Ihrem  Benutzernamen  in  das  System einwaehlen,  da  sonst  Ihr
Mailboxfach wieder geloescht wird.
Teilnehmer,  die durch rechtswidrige  Verwendung der Mailbox auf-
fallen, werden sofort aus dem System entfernt.

Sie  koennen jetzt Ihren Antrag mit dem Befehl SENDE ANTRAG  ver-
schicken. Vergessen Sie nicht, Ihr Passwort mitanzugeben! Es wird
sonst ihr Vorname als Passwort eingetragen.  Die  Bearbeitung des
Antrags erfolgt in der Regel innerhalb von 48 Stunden.
Ist  Ihnen das jetzt alles zu  schnell gegangen,  geben  Sie doch
einfach HILFE USERANTRAG ein!

Bitte in folgenden "Brettern" nachschauen:

AMIGA            (03-11-88)      ANGEBOTE          (12-11-88)
ANREGUNGEN       (04-09-88)      ATARI             (28-10-88)
C-FREAKS         (15-10-88)      CAC-NEWS          (15-09-88)
CCC-AKTUELL      (12-10-88)      COMMODORE         (10-09-88)
DFUE             (11-11-88)      DIVERSES          (12-11-88)
ESSEN&TRINKEN    (20-09-88)      GESUCHE           (12-11-88)
HARDWARE         (08-09-88)      INTERPOOLNET      (07-11-88)
KDV              (02-11-88)      KERMIT            (19-07-88)
KINO             (13-11-88)      LITERATUR         (12-11-88)
MAINFRAME         03-11-88       MEDIA-SIG          25-09-88
MEDIEN           (12-11-88)      MSDOS             (11-11-88)
MUSIKBOX         (12-11-88)      NETWORKS          (28-10-88)
PASCAL           (18-09-88)      POLITIK           (31-10-88)
RECHTSWESEN      (12-10-88)      REISEN            (27-07-88)
RESTAURANT-TIPS  (20-09-88)      SCHLAGZEILEN      (12-10-88)
SCIENCE          (12-10-88)      SOFTWARE          (11-11-88)
STELLENMARKT     (29-08-88)      SYSINFO-D         (10-11-88)
SZENE-KOELN      (07-11-88)      TELEPUBLISHING    (07-11-88)
TELESATZ         (01-07-88)      TERMINE           (05-11-88)
TPUNITS          (08-11-88)      UNIX              (13-08-88)
UNTERHALTUNG     (12-11-88)      VMS               (08-08-88)
WARENTEST        (04-09-88)      WIRTSCHAFT        (09-11-88)
WUG              (14-09-88)

Nr.     Datum Zeit  Abs./Empf.       Kb  Betreff

2115    10-09 15:23 ZENTRALE          1  Gumue: Username ohne Umlaute mit UE !
2225    21-09 22:43 ZENTRALE          1  Jon Webb: Username lautet J.Webb
```

```
Befehl: b *

Liste der "Schwarzen Bretter":

AMIGA            03-11       Informationen rund um den Commodore Amiga
ANGEBOTE         12-11       Angebote der Teilnehmer
ANREGUNGEN       04-09       Verbesserungs- und Brettvorschlaege
ATARI            28-10       Informationen fuer Atari-ST User
C-FREAKS         15-10       Tips, Tricks und Sources zur Programmiersprache C
CAC-NEWS         15-09 #     Neues vom CAC
CCC-AKTUELL      12-10       Aktuelle Informationen des Chaos-Computer-Club
COMMODORE        10-09       Brett fuer Commodore CBM User
DFUE             11-11       Neues aus dem DFUE-Geschehen
DIVERSES         12-11       Diverse Meldungen und Ankuendigungen
ESSEN&TRINKEN    20-09       Kochrezepte, Drinks und Cocktail-Mixturen
GESUCHE          12-11       Gesuche der Teilnehmer
HARDWARE         08-09       Informationen zu neuen Computer-Systemen
INTERPOOLNET     07-11 #     InterPoolNet News / Mailboxlisten
KDV              02-11       Zivildienst-Informationen
KERMIT           19-07       Informationen zu DFUE-Protokollen
KINO             13-11       Kinotips und Filmkritiken
LITERATUR        12-11       Buch-Empfehlungen und Rezensionen
MAINFRAME        03-11 * #   CUG: Large-Systems-Usergroup
MEDIA-SIG        25-09 * #   CUG: Media-SIG Koeln
MEDIEN           12-11       Informationen zu den Medien
MSDOS            11-11       Informationen fuer MS-DOS Benutzer
MUSIKBOX         12-11       LP/CD-Tips und Konzerttermine
NETWORKS         28-10       Internationale Datenbanken und Computernetzwerke
PASCAL           18-09       Problemloesungen in Pascal
POLITIK          31-10       Das politische Diskussionsforum
RECHTSWESEN      12-10       Diskussion allgemeiner Rechtsprobleme
REISEN           27-07       Reisen und Hotel-Empfehlungen
RESTAURANT-TIPS  20-09       Restaurant-Empfehlungen und Gourmet-Tips
SCHLAGZEILEN     12-10       Nachrichten zum Tagesgeschehen
SCIENCE          12-10       Forschung, Technik und Umweltschutz
SOFTWARE         11-11       Algorithmen und Public-Domain-Software
STELLENMARKT     29-08       Arbeitsangebote und Stellengesuche
SYSINFO-D        10-11 #     Systemankuendigungen fuer ProNet
SZENE-KOELN      07-11       Aus den Koelner "Suempfen"
TELEPUBLISHING   07-11       Das Mailbox-Magazin
TELESATZ         01-07 #     Druckvorlagen per Mailbox
TERMINE          05-11       Messetermine, Ankuendigungen und Treffen
TPUNITS          08-11       Turbo-Pascal Version 4.0 Units
UNIX             13-08       Informationen zu UNIX Systemen
UNTERHALTUNG     12-11       Unterhaltsames aus aller Welt
VMS              08-08       Informationen zum VAX/VMS Betriebssystem
WARENTEST        04-09       Verbraucher-Informationen und Warentests
WIRTSCHAFT       09-11       Nachrichten aus dem Wirtschaftsgeschehen
WUG              14-09 * #   Women's Usergroup

Befehl: b kino

Brett KINO: Kinotips und Filmkritiken
Bearbeiter: ELO

Nr.     Datum Zeit  Abs./Empf.      Kb  Betreff

2416    27-10 17:32 ELO              1  "Die Kommissarin"
2417    27-10 17:33 ELO              1  "Falsches Spiel mit Roger Rabbit"
2523    04-11 11:10 ELO              1  "Ich bin Du"
2524    04-11 11:10 ELO              1  "Der Kurier"
2640    10-11 13:52 ELO              1  "Dominick und Eugene"
2641    10-11 13:52 ELO              1  "Die letzte Versuchung Christi"
2711    13-11 02:32 T.WERNER         1  "Killing Blue"

(KINO) Befehl: l 2417 1086

Nachricht Nr.:    2417
Nachricht von:    ELO
Betrifft:         "Falsches Spiel mit Roger Rabbit"
Abgesandt am:     27-10-88, 17:33:44
Empf{nger:        KINO
Text:

# #   #   #  # # #   #  ##
# ## ##   # #  # ##  # #  #
# # # #   ##   # # # # #  #
# #   #   # #  # #  ## #  #
# #   #   #  # # #   #  ##  # # # "Falsches Spiel mit Roger Rabbit"

Menschen  und  Zeichentrickfiguren  leben und lieben in Hollywood, bis
der    Zeichentrickhase    Roger   Rabbit   seine   Frau   von   einem
heruntergekommenen Detektiv bespitzeln 1{~t. Dieser merkt bald, da~ es
um viel mehr geht als nur um Eifersucht. Der Fall ist gef{hrlich...
Mit  dieser  am}santen  Mischung  aus  Zeichentrick-  und Realfilm ist
Robert  Zemecki ein Publikumsfilm gelungen, der Witz, Unterhaltung und
rasantes Tempo zeigt. Auch Fans alter Cartoons kommen auf ihre Kosten.

                                                              M. K¦hler

.. aus dem WDR-Videotext Tafel 545/546 ... von ... Christian .. ELO ..

Nachricht Nr.:    1086
Nachricht von:    ELO
Betrifft:         Nuts... Durchgedreht
Abgesandt am:     23-03-88, 16:21:02
Empf{nger:        KINO
Text:

# #   #   #  # # #   #  ##
# ## ##   # #  # ##  # #  #
# # # #   ##   # # # # #  #
# #   #   # #  # #  ## #  #
# #   #   #  # # #   #  ##  # # # "Nuts... Durchgedreht"

In  Notwehr  t¦tet  ein  Luxus-Callgirl  ihren Freier.  Vor Beginn des
eingentlichen Prozesses will man sie in  die  Psychiatrie  abschieben.
Mit  Unterst}tzung  ihres Anwalts beweist sie ihre Noskplit{t",  wird
zur Ankl{gerin gegen  scheinheilige  B}rgerlichkeit  und  ein  rigides
Rechtssystem.
Martin Ritts Psycho-Drama fesselt bis zur letzten Minute.  Das Res}mee
eines   durch   die   Kindheit  verpfuschten  Lebens  mit  all  seinen
Konsequenzen geht unter die Haut.  Barbra Streisand  spielt  sich  die
verzweifelte  Seele  aus dem Leib und auch Richard Dreyfuss zieht alle
Register der Schauspielkunst.

.. aus dem WDR-Videotext Tafel 610/611 ... von ... Christian .. ELO ..

(KINO) Befehl: ?

Brett            Dialog           Ende             Hilfe
Inhalt           Senden           Umlaut

(KINO) Befehl: e

Befehl: e

Anruf um 11:13:50 beendet.
Auf Wiedersehen, Mailbox-Gast.
```

Abb.5: Hier der Mitschnitt einer Verbindung mit der ProNet-Mailbox

Mailboxen gibt es viele. Weltweit. Sie unterscheiden sich neben dem unterschiedlichen Aufbau und Informationsangebot auch in private und kommerzielle Mailboxen.

...und Boxen

Kommerzielle Mailboxen, z.B. die Telebox der DBP oder das GeoNet, werden hauptsächlich von Firmen zum schnellen Nachrichtenaustausch genutzt. Für den (armen) Privatmann haben sie den Nachteil, daß sie recht teuer sind, so bezahlt man etwa 20 DM pro Monat Grundgebühr und für jede Verbindungsminute und das Versenden von Nachrichten noch einmal etwas. Die privaten Mailboxen werden von Privatpersonen hobbymäßig betrieben und sind in der Regel fast kostenlos. Hier hat man natürlich keine Sicherheit dafür, daß die Daten in falsche Hände geraten oder die Box von heute auf morgen 'offline' geht, also sang- und klanglos den Betrieb einstellt. Wenn man das ganze aber "nur" als schönes Hobby betrachtet, reicht eine "Privatmailbox" bei weitem aus.
Um das zum Teil "erschlagende" Informationsangebot einmal kennenzulernen, finden Sie in unserem Artikel ein paar Mitschnitte aus Mailboxen, in denen der Autor selber aktiv ist. (Die Auswahl ist selbstverständlich rein zufällig und sagt nichts über die "Güte" der Mailbox aus.)
Als erstes wollen wir einmal einen Blick in die KomCom-Mailbox des WDR werfen, die, wie fast alle Mailboxen, in verschiedene Schwarze Bretter und ein persönliches Postfach aufgeteilt ist. Nach Anwahl der Telefonnummer (0221) 210515 meldet sich der Rechner wie in Abbildung 4 zu sehen ist.
Wie Sie anhand der Vielzahl der Schwarzen Bretter sehen können, bietet die Mailbox einen "Haufen" an Informationen. In der "Szene" ist es die bekannteste Mailbox überhaupt. Freilich hat diese "Bekanntheit" auch einen entscheidenden Nachteil: Es ist fast unmöglich, einmal eine freie Leitung zu erwischen...
Als zweite Mailbox soll die ProNet (0221) 249508 vorgestellt werden. Sie läuft auf einem AT-Kompatiblen mit 80-MByte-Festplatte und ist von der Benutzeroberfläche ähnlich wie das KomCom aufgebaut. Auch hier ist das Informationsangebot "erschlagend", wie man im Protokoll in Abbildung 5 sehen kann.
Weniger erschlagend, aber dennoch enorm, ist das Angebot der CTBBS. Diese Mailbox ist unter (0221) 8903821 zu erreichen und läuft als einzig bekannte auf einem CPC464 mit Speichererweiterung, vier 720-kByte-Diskettenlaufwerken und einem PD-Mailboxprogramm. Dementsprechend sieht natürlich auch das Softwareangebot aus (Abb. 6). Wie Sie unschwer feststellen konnten, unterscheidet sich der Aufbau dieser Mailbox von den beiden anderen. Es handelt sich dabei um ein sogenanntes BBS (Bulletin Board System). Diese Form ist vor allen Dingen in den USA sehr häufig anzutreffen.

Programmiert

Bis jetzt haben wir eigentlich alles besprochen, was für das Verständnis der Datenfernübertragung wichtig ist, nur eins fehlt noch: ein Kommunikationsprogramm. Das ist ein Programm, mit dem der CPC auf die Stufe eines Terminals degradiert wird; er dient dann nämlich ausschließlich zur Ein- und Ausgabe von Daten. Mitunter gibt es allerdings sehr komfortable Kommunikationsprogramme, mit denen auch Dateien übertragen werden können...
Die Frage ist nur, welches Programm ist das richtige für mich? Um es gleich vorwegzunehmen, es gibt für die CPCs nicht gerade sehr viele Programme auf dem Markt. Zum Teil sind sie schon bei der RS232C dabei, zum Teil muß man sie sich auch als 'Einzelstück' kaufen (oder selber programmieren...). Bei 'Separat'-Programmen sollte man aber schon

```
Moechtest Du ANSI-Grafik [j/N] ? >

*** C.T.B.B.S. ***** COLOGNE TURBO BULLETIN BOARD SYSTEM ***** C.T.B.B.S. ***

                           300/1200 Baud

                  8 Datenbits, 1 Stopbit, No Parity

                         24 Stunden Online

                          Sysop: Lothar
                        CoSysop: Berthold
                        CoSysop: Guenther


          Alle bisherigen Benutzer dieses Systems, die
          als "nicht gefunden" gemeldet werden, moechte ich
          bitten, sich nochmals einzutragen.


Vorname  > MARCO
Zuname   > PRIES
Passwort > ........

Login               : 02:08  Sonntag 13-Nov-88
Eingetr. Benutzer   : MARCO PRIES

Anrufer Nummer      : 707
Access Zeit heute   : 1
Access Zeit total   : 41
Letzter Anruf       : 02:06  Sonntag 13-Nov-88
Zul. geles. Brief   : 114
Uploads bis heute   : 0
Downloads bis heute: 0

C.T.B.B.S. - Login -

Die C.T.B.B.S. laeuft nun nochmal mit einer neuen Software. Wir wuenschen
allen Usern viel Freude an den neuen, erweiterten Moeglichkeiten. Bei
Problemen schreibt Bitte eine Nachricht an den Sysop.

Druecke...

  "?" fuer Menues und Hilfe                 CTRL-S fuer Listings-Pause
  Zeichen in "<>" sind Befehle              CTRL-C fuer Listing-Abbruch

Die erste Zahl im System-Prompt zeigt die verbleibende System-Zeit an.

Davon oeffentliche Briefe     : 13

In diesem Brett sind keine privaten Briefe fuer Dich.



68-MessageSystem POST [? = Menue]> F

   Dateibereich: LOGIN   Files: 6   Belegt: 31K
AUSZUG  .TQT   2K ¦ ML2400  .TQT   2K ¦ SYSTEM  .DIR   4K ¦
CTBBSUSR.DQC  17K ¦ PCNET   .MQL   3K ¦ TOILETTE.TQT   3K ¦

0 Uploads, 0 Downloads bis heute.

68-FileSystem LOGIN [? = Menue]> DIR

C.T.B.B.S.  FileSystem Menue
=============================================================================
```

```
<B>rett/Dateibereich wechseln              <F>inde Datei(en)
<I>nhalt des Dateibereichs                 <G>roesse/Transferzeit anzeigen
<A>RC-File an/aus                          <N>eue Uploads anzeigen
<L>BR-File an/aus                          <Z>eigen einer ASCII- oder *.?Q?-Datei

<S>ende File mit X-/Y-Modem (Upload)            (Benutzer --> CTBBS)
<H>ole File mit X-/Y-Modem (Download)           (CTBBS --> Benutzer)

Die anderen Menues:           <M>essagesystem          <U>tilities

<E>nde (logoff)


67-FileSystem LOGIN [? = Menue]> B CPM


   Dateibereich: CPM   Files: 13   Belegt: 143K
AUTO     .LBR   4K ¦ KERMIT  .LBR  19K ¦ SWEEP   .COM  12K ¦ USERCOPY.COM   3K
CRUNCH   .LBR  13K ¦ MACTOOLS.ARC  45K ¦ TTT     .COM   6K ¦
CUBE     .LBR  10K ¦ NEWLU   .CQM  15K ¦ UKM7    .COM   8K ¦
INHALT   .LBR   5K ¦ RUN     .BAS   2K ¦ UNERA   .COM   1K ¦

67-FileSystem CPM [? = Menue]> M

67-MessageSystem POST [? = Menue]> L

Lesen Such Option <A><E><G><N><S><Q><?> > A

Oeffentlicher Brief #114 [POST] vom: 16:51  Sonntag  9-Okt-88
  Von: BERTHOLD BOLLINGER
   An: ALLE
Betr.: Nullen nach CR...

Lesen <J><N><F><Q><V><A><?> > J

Zur Information fuer alle, die (wie ich) gelegentlich etwas "verbogene"
Menues auf dem Bildschirm haben:

Dieses Mailboxprogramm verfuegt ueber (fuer einen Z80-Rechner) recht
schnelle I/O-Routinen. Wer einen relativ langsamen Rechner als
DFUE-Terminal verwendet, kann unter Umstaenden erleben, dass am
Zeilenanfang manchmal Zeichen verlorengehen und die Menues dadurch
entstellt oder schlecht leserlich werden.
Abhilfe: Im Utilities-System den Menuepunkt <P>arameter anwaehlen und
die Anzahl der <N>ullbytes nach CR/LF erhoehen (z.B. auf 5-6). Die Box
sendet dann nach jedem CR und LF die entsprechende Anzahl an
Nullzeichen (chr$(0)), und der Terminalrechner hat Zeit genug seinen
Zeilenvorschub zu machen...

Gruss, Berthold.


65-MessageSystem POST [? = Menue]> P

C.T.B.B.S.  MessageSystem Menue
===============================================================================
Briefe:                                        <T>exte zu besonderen Themen
<L>esen          <S>chreiben                   <M>itteilungen des SysOp
<B>rett/Briefbereich wechseln                  <C>AC-Mailboxliste
<K>urzuebersicht Briefe (nur Betreff)          S<Y>steminfo
<I>nhalt Message-System (komplett)

Die anderen Menues:           <F>ilesystem          <U>tilities

<E>nde (logoff)
65-MessageSystem POST [? = Menue]> U
```

CPC-ANWENDUNG

ConText CPC

Autor: Matthias Uphoff

Damit das Schreiben wieder Spaß macht

Der Klassiker:

ConText CPC – bis heute ungeschlagen in der Gruppe der Textverarbeitungen. Dieses Programm besticht vor allem durch seine leichte Anwendungsweise, die selbst unerfahrenen Computerbesitzern den Umgang mit einer Textverarbeitung möglich macht.

ConText CPC – das heißt:

Einfachste Bedienung durch logische Tastaturbelegung; alle Funktionen sind über die Control- und eine definierte Taste zu erreichen.
Funktionen wie EINFÜGEN, FLIESSTEXT, BLOCKFORMATIERUNG und KOPIEREN sind über Tastendruck aufrufbar und werden in einer Statuszeile angezeigt.
Mehrspaltiges Schreiben und Textkopieren erleichtern Ihnen die Korrespondenz.
25 KByte Textspeicher mit insgesamt 5 DIN-A4-Seiten, damit Sie auch lange Briefe problemlos erstellen können.
Voreingestellt für die meisten CENTRONICS-Drucker, durch übersichtliche Tabellenprogrammierung anpaßbar an viele EPSON-kompatible Drucker.
Weiterschreiben während des Druckens, denn 'time is money'.
Darstellung von Sonderschriften wie VERGRÖSSERN und UNTERSTREICHEN, Anzeigen von anderen Schriftarten durch Steuerzeichen.
Auf Diskette / Kassette gespeicherte Textbausteine sind überall im Text plazierbar.
Eingebauter Taschenrechner und Kalender, damit Sie den Überblick behalten.
Dies sind nur einige der Möglichkeiten, die ConText Ihnen als Textverarbeitung bietet.

Der benötigte Hardware-Aufwand ist gering.

Sie brauchen nur einen CPC 464 / 664 oder 6128 und einen Drucker. Alles andere erklärt Ihnen die ausführliche deutsche Bedienungsanleitung, die dem Programm beiliegt.

Und wo gibt's ConText CPC ?
Bei DMV zu bestellen als

3"-Diskette	59,– DM*
Kassette	49,– DM*

Turbo-DATA-CPC

Volldampf in der Dateiverwaltung

Brauchen Sie ein neues Adreßbuch oder Telefonverzeichnis? Müssen Sie Ihre Sammlungen von Briefmarken, Schallplatten, Dias oder anderen Schätzen sortieren? Alles kein Problem, wenn Sie für Ihren CPC eine universelle Dateiverwaltung haben, und die gibt's jetzt beim DMV:

Turbo-DATA-CPC ist die universelle Dateiverwaltung für jede Gelegenheit, mit der Sie problemlos Überblick und Ordnung in alle Ihre Daten bringen können. Dabei helfen Ihnen die vielfältigen Funktionen von Turbo-DATA-CPC:

- Universell durch veränderbare Ein- und Ausgabemasken
- Blitzschnelle Suchfunktionen durch Indexfelder
- Dateikapazität bis maximal 80 verschiedene Felder
- Bis zu 19 Felder gleichzeitig auf dem Bildschirm sichtbar
- Besonders schnelle und umfangreiche Sortierfunktion
- Eigene Formatierroutine mit extrem hoher Kapazität
- Zweiteilung des Bildschirms in Status- und Arbeitsfeld
- Deutscher Zeichensatz und deutsche Tastaturbelegung
- Komplette Druckroutinen, auch für Etikettendruck

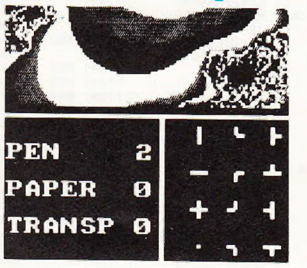
Hardwarevoraussetzung: CPC 464/664/6128 mit einem Diskettenlaufwerk

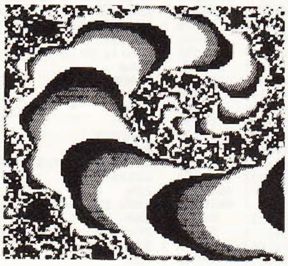
Ganz gleich, was Sie zu archivieren haben, Turbo-DATA-CPC ist in jedem Fall die richtige Wahl.

DM 69,–*

Copyshop

Das universelle Hardcopy-Programm für CPC 464/664/6128
Autor: Matthias Uphoff





Copyshop im Detail:
– Hardcopy in 4 (!) Formaten: DIN A4, DIN A5, 13,5 x 8,5 cm und 21,5 x 13,5 cm – superschnelle Hardcopy-Routine: DIN A4 in ca. 4 Minuten (DMP 4000) – arbeitet in allen drei Modes – Anpaßmenü für JEDEN Epson-kompatiblen Drucker – läuft ebenfalls mit den Seikosha-Druckern GP-500 CPC, GP-550 CPC und GP-1000 CPC – Anpassung an Drucker möglich, die mit 1280 Punkten pro Zeile arbeiten, z.B. CPA-80 GS – Okimate ML 182 – Anpassung kann beim Verlag angefordert werden – Anpassung auch für Drucker möglich, die die Bitbild-Bytes verkehrt herum drucken (das MSB unten statt oben), z.B. NEC P2-Pinwriter. – 32 Farbraster über Menü wählbar – Grafikeditor – komfortable Pulldown-Menüs – Schnelle Fill-Routine – beliebige Ausschnittvergrößerungen – Bildschirm invertieren – selbstrelozierende Hardcopy-Routinen für eigene Programme – neue Save- und Load-Routinen erkennen automatisch Mode und Farbwerte – Freezer – saved auf Tastendruck Screenshots aus laufenden BASIC-Programmen, die anschließend ausgedruckt werden können.

Und die Weltneuheit: **Hardcopy-Simulation auf dem Bildschirm!**
Sie können sich Ihre Hardcopy vor dem endgültigen Ausdruck auf dem Bildschirm ansehen!

3"-Diskette 49,– DM*

SPECIAL OFFERs!

für CPC 464-664-6128, nur auf 3"- Disketten

Original CPC-Software im Paket zu stark herabgesetzten Preisen

DISKSORT-STAR
Leistungsstarke Diskettenverwaltung, die keinem CPC-Benutzer fehlen sollte. DISKSORT-STAR verwaltet, archiviert, katalogisiert, druckt, ... Ihre Diskettensammlung auf einfachste Weise. Neben der reinen Diskettenverwaltung ist unter anderem noch ein kompletter Diskettenmanager enthalten. Auch in puncto Bedienungskomfort ist DISKSORT-STAR kaum zu schlagen.

STAR-MON
Das Entwicklungssystem für Profis
• Assembler • Editor • Disassembler • Monitor • vier Breakpoints • Trace-Funktion • Bankswitch • Memory Dump • Diskettenmonitor • u.v.m.

CPC SPECIAL OFFER No. 2 Vier Programme Best.-Nr. 205

CREATOR-STAR
Ein Trickfilmdesigner für alle Hobby-regisseure auf dem CPC!
• Sprite-Designer • Laufschrift • Utilities • Kulissendesigner • Sprites mit 4 Unterpositionen • Verbinden von Sprites • Kulissen auch übereinandergelegt • Eigene Programmiersprache mit Editor und Compiler

DM 59,–*

DESIGNER-STAR
Grafikprogramm, mit dem man Bildschirmgrafiken komfortabel erstellen kann. Hilfsmenü auf Tastendruck – kein Joystick oder Maus notwendig.

* Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

auf den Preis achten. So gibt es z.B. ein Public-Domain-Programm, Modem Executive (MEX), welches klare Vorteile bietet (der Autor arbeitet selbst damit): Erstens ist es kostenlos, weil eben Public Domain, und zweitens ist es direkt für den Einsatz auf den CPCs vorgesehen (es läuft auf jedem CPC unter CP/M 2.2 mit dem im Handel erhältlichen Amstrad-Interface). Des weiteren unterstützt es die Modem-Funktionen wie automatische Wahl, Telefonnummernregister, XModem, YModem, Online-Help usw.

Glossar

RS 232 C
Amerikanische Schnittstellennorm. Sie beinhaltet:
a) die funktionellen Anforderungen an die Schnittstelle zwischen DEE und DÜE sowie
b) die elektrischen Eigenschaften der Schnittstellenleitungen.
a) DIN 66 020 Teil 1 (CCITTEmpfehlung V.24)
b) DIN 66 259 Teil 1 (CCITTEmpfehlung V.28)

DEE
Datenendeinrichtung. Das ist das Datenendgerät bzw. der Rechner. Die englische Bezeichnung ist DTE (Data Terminal Equipment).

DÜE
Datenübertragungseinrichtung. Das ist das Modem. Die englische Bezeichnung ist DCE (Data Communication Equipment).

CCITT
Comite Consultatif International Telegraphique et Telefonique. Das ist ein internationaler, beratender Ausschuß für Standardisierungen im Übertragungsdienst.

UART
Englische Abkürzung für Universal Asynchronous Receiver Transmitter. Das ist der eigentliche Umwandler innerhalb der seriellen Schnittstelle. Beispiele für einen UART sind unter anderem die Z80 SIO, die Z80 STI oder der Z80 DART.

Modem
Ein Kunstwort aus MOdulator und DEModulator. Ein Modem hat die Aufgabe, digitale Daten (vom Rechner) in analoge umzuwandeln, um sie als Tonfrequenzen in das Telefonnetz zu schicken. Auf der anderen Seite wandelt es die Tonfrequenzen wieder zurück in digitale Signale. Vereinfacht kann man sagen, daß jedem logischen Wert eine bestimmte Frequenz zugeordnet ist.

Simplex
Bezeichnung für eine Übertragungsart, bei der die Daten nur in eine einzige Richtung übertragen werden.

Halbduplex
Bezeichnung für eine Übertragungsart, bei der die Daten abwechselnd von der einen in die andere Richtung übertragen werden.

Duplex
Bezeichnung für eine Übertragungsart, bei der die Daten gleichzeitig in beide Richtungen übertragen werden.

XModem, YModem
Das sind Übertragungsprotokolle, mit denen Daten (nicht nur Texte) gesichert übertragen werden können. Diese Protokolle erkennen weitgehend mögliche Übertragungsfehler und korrigieren diese auch. Die ungesicherte Übertragung bietet indes keine Fehlererkennung oder gar Korrektur.

Literaturverzeichnis

c't 12/86,
Die RS232C-Schnittstelle,
Heise Verlag

DIN 66 020 Teil 1,
CCITT-Empfehlung V.24,
Beuth Verlag

DIN 66 259 Teil 1,
CCITT-Empfehlung V.28,
Beuth Verlag

(Marco Pries/jb)

```
64-Utilities [? = Menue]> P
Parameter <B><C><L><N><P><Q><S><T><?> > ?

                    Aendern der User Parameter

<B> Prompt Bell an/aus    <L> Zeilen per Seite     <P>asswort aendern
<N>ullen nach C/R         <S>hift nock an/aus      <T>ransferprotckoll

<Q>uit/Ende

Alle Parameter werden abgespeichert fuer spaetere Anrufe.

Parameter <B><C><L><N><P><Q><S><T><?> > L
Gegenwaertige Einstellung Lines per Screen ist 23.
Neue Einstellung [10-48 oder 99 = ohne Pausen]> 99
Parameter <B><C><L><N><P><Q><S><T><?> > N
Gegenwaertige Einstellung: 2 Nullen.
Wieviele Nullen benoetigst Du [0-9]? > 0
Parameter <B><C><L><N><P><Q><S><T><?> > Q

63-Utilities [? = Menue]> M

63-MessageSystem POST [? = Menue]> C

 MAILBOXEN KOELN (C) CAC e.V. Verein fuer TeleKommuniKation Vers.01.11.88/11.
-----+--------+-----------------------+--------------------+---+-----+-------
VWahl!  TelNr. ! MailBoxen 24 H Online ! System Info        !PRG! NET ! SYSOP
-----+--------+-----------------------+--------------------+---+-----+-------
0221 !  210515 ! KomKom Multi User     ! WDR Computer Club!CPM!       ! Cristia
0221 !  244054 ! Silly  MBox           ! Atari Download    !AST! Zer ! Erwin
0221 !  361634 ! Senil                 ! MagicNet Koeln    !IBM! MAG ! Gerd
0221 !  394976 ! Cream II Cologne      ! ST Info+SoftWare  !AST!     ! Gerd
0221 !  426760 ! Sunil                 ! BHP CAC CCC IPN   !IBM! PCN ! Georgie
0221 !  511087 ! Fool                  ! Foolish POWER     !IBM! PCN ! Fool
0221 !  512640 ! Hacker Box Koeln      ! CAC/HARDWARE-LAB  !IBM! PCN ! Hans
0221 !  558336 ! BITDschungel LINKS-K  ! CAC/LINKS/CCC/BHP!IBM! ZER ! Peter
0221 !  621278 ! Linex                 ! VKM Box           !AMI!     ! Wolfgan
0221 !  626327 ! DW-Medienservice      ! DatenBank      (K)!IBM!     ! Dwbox
0221 !  693767 ! Nomex - RBBS          ! Clipper/DBase     !IBM!     ! Josef
0221 !  810323 ! PMM - Dbase/Clipper   ! Keine 300 Baud    !IBM! PCN ! Peter
0221 !  856888 ! Kibo                  ! VKM Box           !AMI!     ! Reiner
0221 !  882898 ! MILKA                 ! CAC/Meinungen     !IBM! PCN ! Gerd
0221 !  896323 ! Macro                 ! Tus Info          !AMI!     ! Norbert
0221 ! 1616284 ! Saturn MailBox Koeln  ! Keine 300 Baud(K)!IBM! PCN ! Peter
0221 ! 7393515 ! FOTO-BOX              ! Foto,IBM,Infos    !IBM! PCN ! Elmar
0221 ! 8903821 ! CTBBS                 ! PICS VERS. 1.6 d !CPC!     ! Lothar
0214 !   45834 ! LOS Leverkusen Online ! MagicNet Server   !IBM! MAG ! Roland
02173!   31748 ! CBM                   ! Jetzt 24 Stunden !AMI!     ! Michael
02174!    3371 ! Bobi 1 Burscheid      ! ElectricCircus    !AST!     ! Bob
02203!   24448 ! DISK                  ! 18 h bis 07 h     !IBM! PCN ! Jacky
02203!   25838 ! ABC                   ! FreeWareLine      !AST! ZER ! Uwe
02203!   26860 ! Dataserv              ! SystemBeratung(K)!IBM!     ! Berretz
02203!   33021 ! PAL                   ! Software Line     !IBM! PCN ! Dirk
02203!   33133 ! CREATIV MailBox 0     ! CAC/IPN/APPLE     !IBM!     ! Gregor
02204!   57102 ! PRO NET 1 - CAC -     ! ProNet-Cologne    !IBM!     ! Boris
02234!   63321 ! EKM Koenigsdorf       ! VKM - Box         !C64!     ! Jan
02234!   71292 ! Plakat Box Koeln      ! RestaurantInfo    !IBM!     ! Ralf
02236!   63371 ! Magic Mountain        ! Endlich wieder da!AST! ZER ! Jochen
02236!   81924 ! Arena                 ! Unbekannter HASS !128!     ! Flori
02237!    8459 ! HartBox               ! Kerpen - Brueggen!IBM! PCN ! Bernd
02273!    2637 ! Camel                 ! Spiele Box        !IBM! PCN ! Horst
02273!   55782 ! Fantasy               ! Das Paradis       !AMI!     ! Sammy
-----+--------+-----------------------+--------------------+---+-----+-------
VWahl!  TelNr. ! MailBoxen 24 H Online ! System Info        !PRG! NET ! SYSOP
-----+--------+-----------------------+--------------------+---+-----+-------
02051!   21568 ! VOPATEPATU            ! Velbert           !AST! ZER ! Peter
```

```
0201 !  256885 ! ELOI's Mailbox System ! Essen              !AST! ZER ! Frank
0201 !  627018 ! Milliways             ! Essen              !IBM! MAG ! Patrick
0201 !  790957 ! Megabox               ! Essen              !IBM! MAG ! Arnold
0202 !  463678 ! Ronsdorfer Box        ! Wuppertal          !IBM! ZER ! Horst
0202 !  464259 ! MSWS                  ! WUPpertal          !IBM! PCN ! Axel
0202 !  473086 ! ToelleTurm            ! Wuppertal          !IBM! ZER ! Kerstin
0202 !  597645 ! Wuppertaler Daten Box ! FastNet-Control    !AMI!     ! Klaus
0202 !  661051 ! Wuppertaler Muell Syst! Wuppertal          !C64!     ! Powerb
0202 !  789207 ! Vohwinkler Mailbox    ! Wuppertal          !C64!     ! Knox
0202 ! 4660234 ! Underground Network   ! LAN/FastNet        !IBM!     ! Volker
02151!  476567 ! KDWDB Krefeld /BTX/CAC! Krefeld CAC        !IBM!     ! Karl
02235!   43799 ! AMIGA BOX Erftstadt   ! 20-07 Uhr          !AMI!     ! L-B-A
02241!  404403 ! Midimail              ! Troisdorf          !AST! ZER ! Rene
02244!    5552 ! Public Mail Service   ! OberPleis          !AMI!     ! Juergi
0228 !  254020 ! Maus                  ! Bonn               !IBM! MAU ! Axel
0231 !  334346 ! FORUM                 ! Dortmund           !IBM! PCN ! Frank
02324!   83368 ! KKB                   ! Hattingen          !IBM! MAG ! Kim
0234 !  412240 ! PENTA                 ! Bochum             !IBM! PCN ! Ralf
02361!   29694 ! Nomex                 ! Recklinghausen     !IBM!     ! Rolf
02361!   67335 ! FANTASY               ! Recklinghausen     !IBM! PCN ! Susi
02407!    18326! KICK                  ! Kohlscheid         !AMI!     ! Uwe
02461!   53606 ! JuelicherInfoService  ! Juelich            !128!     ! Willi
02761!    5900 ! SUESAU                ! Olpe               !IBM!     ! Gregor
030  ! 4642336 ! CPM                   ! AMATEURFUNK        !IBM! PCN ! Chmie
030  ! 6818679 ! Info Box Berlin       ! Berlin/IPN         !IBM!     ! Heino
06201!   66995 ! BMW                   ! Weinheim           !IBM! PCN ! Claus
0631 !   90469 ! PC-INFO               ! Kaiserslautern     !IBM! PCN ! Klaus
06525!    7321 ! EifelBox              ! Irrel/Eifel        !IBM! PCN ! Michael
0681 !   582627! BCB                   ! SAARbruecken       !IBM! PCN ! Benny
0681 !  635480 ! NEBOSA                ! SAARbruecken       !IBM! PCN ! Kurt
069  ! 4990769 ! The Dungeon - Amiga   ! FrankFurt          !AMI!     ! Mad Max
089  ! 5706448 ! LINKS                 ! Muenchen           !IBM! ZER ! Taus
09120!    9939 ! JAT Mailbox           ! Leinburg           !IBM!     ! Achim
0032 ! 87866321! PAL-PDB               ! Belgien            !IBM! PCN ! Ralf
0041 ! 31460001! TRANSFORM             ! Schweiz/Bern       !IBM! PCN ! Bernd
0041 ! 45211488! INVESTRA              ! Schweiz/Kaltbach !IBM! ZER ! Gino
-----+--------+-----------------------+--------------------+---+-----+-------
  MAILBOXEN KOELN (C) CAC e.V. Verein fuer TeleKommuniKation 01.11.88/11.01
 :-----> ---------------------------------------------------------- <-----
 :Info:HARDWARE LAB 0221-515529 HANS ist von 15-22 Uhr zu erreichen:CAC<-----
 :-----> ---------------------------------------------------------- <-----
 :-----> PCNet - PCN - ZERBERUSNET - ZER - MAGICNet MAG - MAUSNET - MAU<-----
 :---------------------------------------------------------------------------
 :-----> USERTREFFEN :Am Samstag dem 12.11.1988 im JugendPark unter der<-----
 :-----> ZooBruecke, SachsenbergStrasse, 5000 Koeln 21 (Deutz) 14-19 h <-----
 :-----> ---------------------------------------------------------- <-----
 :-----> AN ALLE SYSTEMBETREIBER - Bitte sendet Eure WERBEZEILE in die <-----
 :-----> SUNIL - 0221/426760 PCNEt Koeln -  AN ALLE SYSTEMBETREIBER -  <-----
 :-----> ---------------------------------------------------------- <-----
 :-----> CAC e.V.- Verein fuer Telekommunikation - :Eupener Strasse 161<-----
 :-----> 5000 Koeln 41: Tel (Buero): 0221-497 28 83: 10:00 - 18:00 Uhr <-----
 :---------------------------------------------------------------------------
 : (c) CAC e.V. & Georgie : Liste ausdrucken und gebrauchen.1101880111 <-----
  -------------------------------------------------------------------------
 < Freeware by Computer Artists Cologne e.V. - Verein fuer Telekommunikation
 < Nicht fuer kommerziellen Gebrauch (C) Georgie;CAC e.V. "Lass mich REISEN"
   -----------------------------------------------------------------------

59-MessageSystem POST [? = Menue]> E
Hast Du noch einen Kommentar an den SysOp [j/N] ? >

Logoff: 02:19  Sonntag 13-Nov-88
Verbindung dauerte: 11 Min.
Auf Wiedersehen, MARCO! Ruf mal wieder an...
```

Abb.6.: Die CTBBS-Mailbox hat ein etwas anderes Aussehen als die anderen Mailboxen. Näheres erfahren Sie im Text

Fast wie im Fernsehen

Alternativer Bildschirmaufbau mit Screenflutsch

Nehmen wir an, Sie haben ein Programm geschrieben, welches sich zwecks hübscherer Darbietung eines Titelbildes bedient. Nehmen wir weiter an, Sie haben beschlossen, daß das Bild effektvoll erscheinen soll. Mit vorliegendem Programm stellen wir Ihnen eine Möglichkeit vor, dieses Problem zu lösen.

Der Effekt, den Screenflutsch Ihnen bietet, ist ein seichtes Einblenden der Grafiken. Er wird dadurch erreicht, daß vor dem Laden des Bildes alle Farben auf den Wert 0, also schwarz, gesetzt werden. Danach werden die einzelnen Farbstifte der Reihe nach mit Farbzuweisungen bedacht. Die einzelnen Fragmente der Zeichnung, die mit einem Farbstift gemalt wurden, ändern so in vom Benutzer bestimmbaren Abständen ihre Farbe, bis sie das endgültige Aussehen erreicht haben.
Das Programm fragt nach Start einige Variablen ab, durch die der Modus, die Verzögerungszeit und der Name der Bilddatei bestimmt werden.
Sollten Sie diese Routine in eigenen Programmen gebrauchen, so können Sie Zeilen 20-60 weglassen, müssen allerdings die Variablenzuweisungen im Programm vornehmen.
Um das Programm nach dem Abtippen testen zu können, sollte sich ein beliebiges Bild (übliche 17 kByte Binärdatei) auf dem Datenträger befinden, da das Programm ohne derartiges "Futter" selbstverständlich nicht läuft. Wenn Sie allerdings alle Vorbereitungen richtig getroffen haben, ja dann gibt es was zu sehen, eben fast wie im Fernsehen.

(Stefan Westner/jf)

für 464-664-6128

```
10 'Screenflutsch - 1989 by Stefan Westner [1707]
20 MODE 2:PAPER 0:BORDER 0:INK 0,0:INK 1,2 [3430]
5 'Zeile 20-60 nur fuer Demozwecke!
30 INPUT"Welchen Mode [0/1/2]? ";bmode      [3196]
40 INPUT"Welches Bild? ";bild$              [1456]
50 INPUT"Verzoegerung? ";verz               [3076]
60 IF bmode=1 THEN farben=3 ELSE IF bmode= [3826]
2 THEN farben=1 ELSE farben=15
70 MODE bmode:BORDER 0:FOR i=0 TO farben:I [4131]
NK i,0:NEXT 'Alle Farben auf 0 setzen
80 LOAD bild$,&C000 'Bild laden(muss eine  [4375]
17KB-grosse Bildschirmdatei sein)
90 FOR i=0 TO farben          'Schleife von [2824]
 0 bis Farbenanzahl
100    READ c                 'Farbwert les [1290]
en
110    FOR a=0 TO c           'Farbe von 0  [3119]
bis Farbwert durchlaufen
120        INK i,a                          [516]
130        FOR b=0 TO verz    'Wartezeit zw [4734]
ischen einem Farbwechsel
140        NEXT b                           [378]
150    NEXT a                               [383]
160 NEXT i                                  [375]
170 'Farbdaten - Bei Mode 0 sind es 16(0-1 [3836]
5), bei Mode 1 sind es 4(0-3)
180 DATA 26,13,0,15,24,6,16,2,14,1,3,11,10 [2524]
,9,18,25
190 CALL &BB18                              [389]
```

Listing FLUTSCH



Timer – Die Weltzeit im CPC

Eine praktische Anwendung für alle Globetrotter

Sie kennen das Problem: Ihre Geldbörse ist leer, keine Aussicht auf positive Eingänge auf Ihr Girokonto und gähnende Leere in Ihrem Kühlschrank. Sie stürmen zum Telefon und wählen eine beliebige Nummer in den Vereinigten Staaten, in der Hoffnung, der fernmündliche Adressat entpuppt sich als reicher Onkel.

Meist endet eine solche Aktion damit, daß Sie am anderen Ende eine völlig verschlafene und damit nicht gerade freundlich gestimmte Person erreichen, deren mögliche Antwortsfloskeln an dieser Stelle aus ästhetischen Gründen nicht beschrieben werden sollten.

Es dauert meist nicht lange, bis Sie den Grund der Entrüstung Ihres erhofften reichen Onkels erkannt haben: die Zeitverschiebung. Damit solche Zwischenfälle nicht zur Gewohnheit werden, können CPC-Besitzer nun auf eine interessante Anwendung zurückgreifen, die in der Lage ist, die Zeit in jedem nur denkbaren Land der Erde zu berechnen und darüberhinaus als Zugabe den Wochentag eines beliebigen Datums bestimmt.

Vorbereitungen

Bevor Timer das erste Mal benutzt werden kann, müssen fünf kurze Programme gestartet werden, die die notwendigen Daten für das Hauptprogramm erzeugen und in Dateien auf Diskette ablegen. Diese kurzen Programme tragen die Namen NR1.BAS, NR2.BAS, NR3.BAS, NR4.BAS und NR5.BAS.

Es entstehen sieben Dateien, die folgende Namen besitzen:

AFRIKA.DAT
ASIEN.DAT
AUSTRAL.DAT
EUROPA.DAT
KANADA.DAT
NORDAM.DAT
SUDAM.DAT.

Wenn Timer fehlerfrei laufen soll, müssen letztgenannte Dateien nebst Hauptprogramm TIMER.BAS auf Ihrer Diskette vorhanden sein.

Ruft man das Programm auf, so gelangt man in ein Auswahlmenü, in dem die beiden Teilfunktionen von Timer angewählt werden können.

Wochentagbestimmung

Dieser Teil Timers klärt Sie über den Wochentag eines beliebigen Datums auf. Alles, was Sie hier zu tun haben, ist die Eingabe des aktuellen und des gewünschten Datums. Dabei haben Sie das Format tt.mm.jjjj einzuhalten. Der CPC untersucht zuerst, ob das eingegebene Datum auch tatsächlich existiert und bestimmt daraufhin den Wochentag des gesuchten Datums.

Das verlangte aktuelle Datum dient dabei dem Vergleich mit dem gesuchten und sorgt so dafür, daß die Ausgabe auch vom Wortlaut her stimmt. Liegt das gesuchte Datum in der Vergangenheit, so wird das vom CPC vermerkt und so fort.

Weltzeit

Die zweite Funktion des Programms besteht in der exakten Bestimmung der Weltzeit.

In einem Menü kann der Kontinent angewählt werden, auf dem sich das Land befindet, dessen momentane Zeit wir bestimmen wollen. Die geographischen Kenntnisse nimmt einem der Computer hier nicht ab; man sollte schon wissen, daß Brasilien in Südamerika zu finden ist und nicht etwa schräglinks an Indien angrenzt. Nun ist noch die aktuelle Zeit einzugeben, fertig.

Timer greift nun auf die auf Diskette gelagerten Dateien zurück und bestimmt die Zeiten.

Auf dem Bildschirm erscheint nach kurzer Zeit eine Tabelle, der wir unsere gewünschten Informationen entnehmen können. Alle wichtigen Länder des betreffenden Erdteils werden in alphabetischer Folge inklusive der augenblicklichen Uhrzeit aufgelistet.

Da bei einigen Kontinenten die Anzahl der Länder doch erhebliche Dimensionen erreichen, sind diese zweckmäßig in sinnvolle Gruppen aufgeteilt. So erhält man bei der Wahl des amerikanischen Kontinentes zuerst die Länder Kanadas, dann die Nordamerikas und schließlich die Südamerikas präsentiert.

Sind alle vorhandenen Daten an den Benutzer weitergegeben, so hat er die Möglichkeit, das Programm zu beenden oder einen Neustart auszulösen.

Timer ist sehr einfach zu benutzen und bietet eine Menge an Informationen.

Wer etwas firm in der Programmierung des CPCs ist, der kann das gut lesbare Programm nach eigenen Wünschen modifizieren oder auch erweitern. Denkbar ist eine Veränderung der Hilfsprogramme, die alle Daten enthalten und die für Timer wichtigen Dateien auf Diskette schreibt. Will man beispielsweise die Länder Europas um Bayern ergänzen, so ändert man in Zeile 30 des Programms

NR2.BAS die Dimensionierungen auf 26. In Zeile 50 wird die Laufanweisung ebenfalls auf 26 erhöht. Nun fügen wir eine Datazeile ein:
125 DATA Bayern,0
Die 0 steht für die Zeitveränderung.

Da Bayern ja bekanntlich die gleiche Uhrzeit besitzt wie das restliche Europa, ist hier keine Zeitveränderung (also 0) zu vermerken. Schon ist die Ergänzung getätigt und das Programm um eine Information reicher.

Hinweis:
Benutzen Sie einen CPC 464, so müssen Sie im Listing den "FILL"-Befehl herausnehmen, oder Sie benutzen den Emulator (im Heft enthalten).

(Michael Peter/jf/cd)

für 464-664-6128

```
10 ' ********************************** [1391]
20 ' *                                * [175]
30 ' *          T I M E R             * [93]
40 ' *                                * [175]
50 ' *        ein  Programm           * [591]
60 ' *   zur Wochentags-Bestimmung    * [1830]
70 ' *             und                * [189]
80 ' *        Weltzeitanzeige         * [1026]
90 ' *                                * [175]
100 '*             von                * [269]
110 '*                                * [175]
120 '*       MICHAEL  PETER           * [1307]
130 '*                                * [175]
140 '*Copyright 1989 DMV-Verlag V 1.0 * [1719]
150 '********************************** [1391]
160 DIM aa$(100),t(100)                  [752]
170 SYMBOL AFTER 250                     [1398]
180 SYMBOL 250,11,11,11,11,11,11,11,11   [2080]
190 SYMBOL 251,208,208,208,208,208,208,208 [2853]
,208
200 SYMBOL 252,251,99,55,27,13,6,3,1     [2189]
210 SYMBOL 253,223,198,236,216,176,96,192, [2103]
128
220 pfeil$=CHR$(250)+CHR$(251)+CHR$(10)+CH [3860]
R$(8)+CHR$(8)+CHR$(252)+CHR$(253)
230 null$=CHR$(32)+CHR$(32)+CHR$(10)+CHR$( [4992]
8)+CHR$(8)+CHR$(32)+CHR$(32)
240 '                                    [117]
250 ' Menue                              [1057]
260 '                                    [117]
270 RESTORE 290:MODE 1:FOR i=0 TO 3:INK i, [2557]
0:NEXT
280 FOR i=1 TO 2:READ x,y:PLOT x,y,1:FOR i [4799]
i=1 TO 4:READ x,y:DRAW x,y:NEXT ii,i
290 DATA 1,1,1,399,639,399,639,1,1,1,4,4,4 [3007]
,395,635,395,635,4,4,4
300 LOCATE 14,3:PRINT "HAUPT - MENUE":FOR [9095]
i=1 TO 2:READ x,y,a,c:LOCATE x,y:PRINT STR
ING$(a,CHR$(c)):NEXT:FOR i=1 TO 6:READ x,y
,c:LOCATE x,y:PRINT CHR$(c):NEXT
310 DATA 14,4,13,154,14,2,13,154,13,2,150, [4636]
27,2,156,27,4,153,13,4,147,13,3,149,27,3,1
49
320 LOCATE 16,8:PEN 3:PRINT "Wochentag":y1 [4518]
=296:y2=264:x=160:i=1:GOSUB 350
330 LOCATE 16,12:PEN 1:PRINT "Weltzeit":y1 [4263]
=232:y2=200:x=144:i=1:GOSUB 350
340 LOCATE 16,16:PEN 1:PRINT "Ende":y1=168 [3840]
:y2=136:x=80:i=1:GOSUB 350:GOTO 370
350 PLOT 230,y1,i:DRAWR x,0:DRAWR 0,-32:DR [5672]
AWR -x,0:DRAWR 0,32:PLOT 227,y1+4:DRAWR x+
7,0:DRAWR 0,-40:DRAWR -(x+7),0:DRAWR 0,40
360 RETURN                               [555]
370 PLOT 217,310,1:DRAWR 187,0:DRAWR 0,-18 [7680]
5:DRAWR -187,0:DRAWR 0,185:PLOT 213,314:DR
AWR 195,0:DRAWR 0,-193:DRAWR -195,0:DRAWR
0,193
380 MOVE 233,y1+10:FILL 2:y1=232:y2=200:x= [6145]
144:i=0:GOSUB 350:y1=168:y2=136:x=80:i=0:G
OSUB 350
390 LOCATE 10,22:PRINT "Cursortasten + <EN [2806]
TER>"
400 FOR i=1 TO 6:READ x,y,c:LOCATE x,y:PRI [3979]
NT CHR$(c):NEXT
410 DATA 9,21,150,32,21,156,9,23,147,32,23 [3145]
,153,9,22,149,32,22,149
420 LOCATE 10,21:PRINT STRING$(22,CHR$(154 [3982]
)):LOCATE 10,23:PRINT STRING$(22,CHR$(154)
)
430 INK 0,0:INK 1,26:INK 2,13:INK 3,10,20: [1988]
BORDER 10:SPEED INK 2,2
440 t=1                                  [276]
```

Listing TIMER

```
450 CALL &BB18                           [389]
460 IF INKEY(2)=0 THEN 480 ELSE IF INKEY(1 [3091]
8)=0 THEN 540 ELSE 450
470 GOTO 450                             [442]
480 IF t=1 THEN t=2:PEN 1:LOCATE 16,8::PRI [11407]
NT "Wochentag":y1=296:y2=264:x=160:i=0:GOS
UB 350:LOCATE 16,12:PEN 3:PRINT "Weltzeit"
:y1=232:y2=200:x=144:i=1:GOSUB 350 ELSE 50
0
490 GOTO 450                             [442]
500 IF t=2 THEN t=3:PEN 1:LOCATE 16,12:PRI [8526]
NT "Weltzeit":y1=232:y2=200:x=144:i=0:GOSU
B 350:LOCATE 16,16:PEN 3:PRINT "Ende":y1=1
68:y2=136:x=80:i=1:GOSUB 350 ELSE 520
510 GOTO 450                             [442]
520 IF t=3 THEN t=1:PEN 1:LOCATE 16,16:PRI [10236]
NT "Ende":y1=168:y2=136:x=80:i=0:GOSUB 350
:LOCATE 16,8:PEN 3:PRINT "Wochentag":y1=29
6:y2=264:x=160:i=1:GOSUB 350 ELSE 450
530 GOTO 450                             [442]
540 IF t=1 THEN 550 ELSE IF t=2 THEN 1210 [2443]
ELSE IF t=3 THEN CALL 0 ELSE 450
550 '                                    [117]
560 ' Wochentag                          [1054]
570 '                                    [117]
580 MODE 1:PEN 1                         [1036]
590 LOCATE 7,1:PRINT CHR$(150);STRING$(25, [9063]
CHR$(154));CHR$(156):LOCATE 7,2:PRINT CHR$
(149):LOCATE 33,2:PRINT CHR$(149):LOCATE 7
,3:PRINT CHR$(147);STRING$(25,CHR$(154));C
HR$(153)
600 LOCATE 9,2:PRINT "Wochentags - Berechn [3670]
ung"
610 '                                    [117]
620 '    AKTUELLES DATUM                 [1210]
630 '                                    [117]
640 LOCATE 6,5:PRINT "Bitte aktuelles Datu [5584]
m eingeben"
650 INK 3,12:PLOT 80,312,3:DRAWR 464,0:DRA [4659]
WR -232,-30:DRAWR -232,30:MOVE 105,310:FIL
L 3
660 PRINT CHR$(2):LOCATE 20,9:INPUT dat$:T [4273]
AG:MOVE 232,272:PEN 1:PRINT dat$;"      "
;:TAGOFF
665 kont$=dat$:fehler=0:GOSUB 691:IF fehle [4060]
r=1 THEN 580
670 '                                    [117]
680 '   FEHLEINGABE                      [1089]
689 '                                    [117]
690 GOTO 700                             [425]
691 IF ((MID$(kont$,1,2)="30") OR (MID$(ko [4541]
nt$,1,2)="31")) AND MID$(kont$,4,2)="02" T
HEN fehler=1
692 IF MID$(kont$,1,2)="31" AND MID$(kont$ [2834]
,4,2)="04" THEN fehler=1
693 IF MID$(kont$,1,2)="31" AND MID$(kont$ [2426]
,4,2)="06" THEN fehler=1
694 IF MID$(kont$,1,2)="31" AND MID$(kont$ [2826]
,4,2)="09" THEN fehler=1
695 IF MID$(kont$,1,2)="31" AND MID$(kont$ [2599]
,4,2)="11" THEN fehler=1
696 RETURN                               [555]
700 IF LEN(dat$)<>10 THEN GOSUB 710:GOTO 6 [9123]
60 ELSE TAG:MOVE 232,272:PRINT STRING$(10,
" ");:TAGOFF:LOCATE 6,5:PRINT STRING$(30,"
 "):GOTO 750
710 LOCATE 1,20:PRINT "  Bitte das Datum v [14529]
ollstaendig angeben.":LOCATE 11,22:PRINT "
Beispiel: 10.01.1988":FOR i=1 TO 2500:NEXT
:LOCATE 1,20:PRINT STRING$(39," ");:LOCATE
 11,22:PRINT STRING$(20," ");:TAG:MOVE 232
,272:PRINT STRING$(10," ");:TAGOFF:RETURN
720 '                                    [117]
730 '    EINGABE                         [925]
740 '                                    [117]
750 LOCATE 6,5:PRINT "Bitte gesuchtes Datu [4942]
m eingeben"
```

Listing TIMER

```
760 INK 3,12:PLOT 80,312,3:DRAWR 464,0:DRA [4659]
WR -232,-30:DRAWR -232,30:MOVE 105,310:FIL
L 3
770 PRINT CHR$(2):LOCATE 20,9: PRINT"      [10635]
          ":LOCATE 20,9:INPUT datum$:kont$=d
atum$:fehler=0:GOSUB 691:IF fehler=1 THEN
750 ELSE TAG:MOVE 232,272:PEN 1:PRINT datu
m$;"          ";:TAGOFF
780 '                                      [117]
790 '   FEHLEINGABE                        [1089]
800 '                                      [117]
810 IF LEN(datum$)<>10 THEN GOSUB 710:GOTO [10186]
 770 ELSE TAG:MOVE 232,272:PRINT STRING$(1
0," ");:TAGOFF:LOCATE 6,5:PRINT STRING$(30
," "):GOTO 850
820 '                                      [117]
830 '   BERECHNUNG                         [895]
840 '                                      [117]
850 c$=MID$(datum$,7,2):'  CENTURY         [1841]
860 d$=MID$(datum$,9,2):'  YEAR            [1746]
870 t$=MID$(datum$,1,2):'  DAY             [1679]
880 m$=MID$(datum$,4,2):'  MONTH           [362]
890 c=VAL(c$):t=VAL(t$)                    [1808]
900 m=VAL(m$):IF m=1 THEN m=11:GOSUB 1090  [5475]
ELSE IF m=2 THEN m=12:GOSUB 1090 ELSE m=m-
2:d=VAL(d$)
910 w=t+(2*m)+INT(((3*m)-1)/5)+d+INT(d/4)+ [3786]
INT(c/4)-(2*c)
920 day=w MOD 7                            [378]
930 IF day=0 THEN day$="Sonntag" ELSE IF d [16722]
ay=1 THEN day$="Montag" ELSE IF day=2 THEN
 day$="Dienstag" ELSE IF day=3 THEN day$="
Mittwoch" ELSE IF day=4 THEN day$="Donners
tag" ELSE IF day=5 THEN day$="Freitag" ELS
E IF day=6 THEN day$="Samstag"
940 GOSUB 1140                             [823]
950 LOCATE INT((40-(20+LEN(day$)+LEN(a$))) [5876]
/2),12:PRINT "Der ";datum$;" ";a$;" ein ";
day$;"."
960 LOCATE 1,25:PRINT CHR$(24);"    Ende   [4582]
    Weltzeit      Neustart     ";CHR$(24);
970 x=6:x1=6                               [974]
980 LOCATE x1,23:PRINT null$:LOCATE x,23:P [3720]
RINT pfeil$
990 CALL &BB18                             [389]
1000 IF INKEY(1)=0 THEN GOTO 1010 ELSE IF  [5934]
INKEY(8)=0 THEN GOTO 1020 ELSE IF INKEY(18
)=0 THEN GOTO 1030 ELSE 990
1010 IF x=6 THEN x1=x:x=18:GOTO 980 ELSE I [6084]
F x=18 THEN x1=x:x=32:GOTO 980 ELSE IF x=3
2 THEN x1=x:x=32:GOTO 980
1020 IF x=32 THEN x1=x:x=18:GOTO 980 ELSE  [6829]
IF x=18 THEN x1=x:x=6:GOTO 980 ELSE IF x=6
 THEN x1=x:x=6:GOTO 980
1030 IF x=6 THEN CALL 0 ELSE IF x=32 THEN  [2737]
MODE 1:GOTO 270 ELSE 1180
1040 LOCATE 1,25:PRINT STRING$(40," ");:GO [2181]
TO 750
1050 END                                   [110]
1060 '                                     [117]
1070 ' UNTERPROGRAMM GLATTE HUNDERTER      [1631]
1080 '                                     [117]
1090 IF VAL(d$)=0 THEN d=99:c=VAL(c$)-1 EL [3697]
SE d=VAL(d$)-1
1100 RETURN                                [555]
1110 '                                     [117]
1120 ' UNTERPROGRAMM VERB                  [1538]
1130 '                                     [117]
1140 a=VAL(MID$(dat$,7,4)):b=VAL(MID$(datu [8883]
m$,7,4)):IF a>b THEN a$="war":RETURN ELSE
IF a<b THEN a$="wird":RETURN ELSE 1150
1150 a=VAL(MID$(dat$,4,2)):b=VAL(MID$(datu [8515]
m$,4,2)):IF a>b THEN a$="war":RETURN ELSE
IF a<b THEN a$="wird":RETURN ELSE 1160
1160 a=VAL(MID$(dat$,1,2)):b=VAL(MID$(datu [7329]
m$,1,2)):IF a>b THEN a$="war":RETURN ELSE
IF a<b THEN a$="wird":RETURN ELSE a$="ist"
1170 RETURN                                [555]
1180 '                                     [117]
1190 ' Weltzeituhr                         [646]
1200 '                                     [117]
1210 MODE 1:PAPER 2:CLS:INK 3,10,20:PEN 1: [3304]
BORDER 13
1220 WINDOW#1,13,27,7,23:PAPER#1,0:CLS#1   [2595]
1230 LOCATE 12,2:PRINT CHR$(150):LOCATE 13 [4068]
,2:PRINT STRING$(16,CHR$(154)):LOCATE 29,2
:PRINT CHR$(156)
1240 LOCATE 12,3:PRINT CHR$(149);" Weltzei [2817]
t - Uhr ";CHR$(149)
```

Listing TIMER

```
1250 LOCATE 12,4:PRINT CHR$(147):LOCATE 13 [6590]
,4:PRINT STRING$(16,CHR$(154)):LOCATE 29,4
:PRINT CHR$(153)
1260 t=1                                   [276]
1270 PLOT 201,295,1:DRAWR 220,0:DRAWR 0,-2 [7928]
52:DRAWR -220,0:DRAWR 0,252:PLOT 197,299:D
RAWR 228,0:DRAWR 0,-260:DRAWR -228,0:DRAWR
 0,260
1280 LOCATE#1,3,3:PEN#1,3:PRINT#1,"Amerika [5894]
":y1=281:y2=249:x=128:i=1:GOSUB 1340
1290 LOCATE#1,3,6:PEN#1,1:PRINT#1,"Europa" [5476]
::y1=233:y2=201:x=112:i=0:GOSUB 1340
1300 LOCATE#1,3,9:PEN#1,1:PRINT#1,"Afrika" [4569]
:y1=185:y2=153:x=112:i=0:GOSUB 1340
1310 LOCATE#1,3,12:PEN#1,1:PRINT#1,"Asien" [2990]
:y1=137:y2=105:x=96:i=0:GOSUB 1340
1320 LOCATE#1,3,15:PEN#1,1:PRINT#1,"Austra [5007]
lien":y1=89:y2=57:x=192:i=0:GOSUB 1340
1330 LOCATE 11,25:PRINT "BITTE WAEHLEN SIE [2372]
 !":GOTO 1350
1340 PLOT 215,y1,i:DRAWR x,0:DRAWR 0,-32:D [7039]
RAWR -x,0:DRAWR 0,32:PLOT 211,y1+4:DRAWR x
+8,0:DRAWR 0,-40:DRAWR -(x+8),0:DRAWR 0,40
:RETURN
1350 a$=INKEY$:IF a$="" THEN 1350          [1474]
1360 IF INKEY(2)=0 THEN 1380 ELSE IF INKEY [1653]
(18)=0 THEN 1500 ELSE 1350
1370 GOTO 1350                             [460]
1380 IF t=1 THEN t=2:PEN#1,1:LOCATE#1,3,3: [10922]
PRINT#1,"Amerika":y1=281:y2=249:x=128:i=0:
GOSUB 1340:PEN#1,3:LOCATE#1,3,6:PRINT#1,"E
uropa":y1=233:y2=201:x=112:i=1:GOSUB 1340
ELSE 1400
1390 GOTO 1350                             [460]
1400 IF t=2 THEN t=3:PEN#1,1:LOCATE#1,3,6: [10437]
PRINT#1,"Europa":y1=233:y2=201:x=112:i=0:G
OSUB 1340:PEN#1,3:LOCATE#1,3,9:PRINT#1,"Af
rika":y1=185:y2=153:x=112:i=1:GOSUB 1340 E
LSE 1420
1410 GOTO 1350                             [460]
1420 IF t=3 THEN t=4:PEN#1,1:LOCATE#1,3,9: [9431]
PRINT#1,"Afrika":y1=185:y2=153:x=112:i=0:G
OSUB 1340:PEN#1,3:LOCATE#1,3,12:PRINT#1,"A
sien":y1=137:y2=105:x=96:i=1:GOSUB 1340 EL
SE 1440
1430 GOTO 1350                             [460]
1440 IF t=4 THEN t=5:PEN#1,1:LOCATE#1,3,12 [12615]
:PRINT#1,"Asien":y1=137:y2=105:x=96:i=0:GO
SUB 1340:PEN#1,3:LOCATE#1,3,15:PRINT#1,"Au
stralien":y1=89:y2=57:x=192:i=1:GOSUB 1340
 ELSE 1460
1450 GOTO 1350                             [460]
1460 IF t=5 THEN t=1:PEN#1,1:LOCATE#1,3,15 [9821]
:PRINT#1,"Australien":y1=89:y2=57:x=192:i=
0:GOSUB 1340:PEN#1,3:LOCATE#1,3,3:PRINT#1,
"Amerika":y1=281:y2=249:x=128:i=1:GOSUB 13
40 ELSE 1350
1470 GOTO 1350                             [460]
1480 '                                     [117]
1490 '                                     [117]
1500 IF t=1 THEN dat$="Kanada.dat":land$=" [3478]
Kanada":o=12:GOTO 1550 ELSE 1510
1510 IF t=2 THEN dat$="Europa.dat":land$=" [4863]
Europa":o=25:GOTO 1550 ELSE 1520
1520 IF t=3 THEN dat$="Afrika.dat":land$=" [4886]
Afrika":o=43:GOTO 1550 ELSE 1530
1530 IF t=4 THEN dat$="Asien.dat":land$="A [5399]
sien":o=35:GOTO 1550 ELSE 1540
1540 IF t=5 THEN dat$="Austral.dat":land$= [4558]
"Australien":o=12:GOTO 1550 ELSE 1350
1550 MODE 2:BORDER 0:PAPER 0               [1815]
1560 LOCATE 2,1:INPUT "Bitte aktuelle Uhrz [10858]
eit (ss.mm) eingeben oder nur <ENTER> drue
cken";tt$:IF tt$="" THEN tt$="12":t1=12:t2
=0:GOTO 1580
1570 IF tt$>"24.00" OR tt$<"00.00" THEN 15 [5825]
60 ELSE t1=VAL(LEFT$(tt$,2)):t2=VAL(MID$(t
t$,4,2)):tt$=LEFT$(tt$,2)
1580 LOCATE 1,1:PRINT STRING$(80," ");:LOC [5456]
ATE 14,1:PRINT "Wenn es in Europa nach MEZ
 ";t1;":";t2;" ist,dann ist es in"
1590 FOR x=1 TO 80:LOCATE x,4:PRINT CHR$(1 [3537]
54);:NEXT
1600 LOCATE 20,4:PRINT CHR$(158):LOCATE 60 [5831]
,4:PRINT CHR$(158):FOR y=5 TO 20:LOCATE 20
,y:PRINT CHR$(149):LOCATE 60,y:PRINT CHR$(
149):NEXT
1610 a=1:s=1:x1=1:x2=22:x3=23:iu=4         [1245]
1620 OPENIN dat$:LOCATE 2,3:PRINT UPPER$(l [2178]
and$)
```

Listing TIMER

```
1630      FOR i=1 TO o                        [455]
1640        INPUT#9,aa$(i),t(i)               [1436]
1650        t0=t1:t00=t2:t1=t1+INT(t(i)):IF t [5015]
1>24 THEN t1=t1-24 ELSE IF t1<0 THEN t1=24
+t1
1660        at=t(i)-INT(t(i)):at=at*60:t2=t2+ [3525]
at:IF t2>59 THEN t1=t1+1:t2=t2-60
1670        LOCATE x1,i+iu:PRINT aa$(i):IF t1 [3370]
>9 THEN LOCATE x2,i+iu ELSE LOCATE x3,i+iu
1680        IF t2<>0 THEN PRINT t1;":";t2 ELS [2158]
E PRINT t1;": 00"
1690        t1=t0:t2=t00                      [1014]
1700        IF i MOD 48=0 THEN s=2:x1=41:x2=6 [3230]
2:x3=63:iu=-44:GOTO 1730 ELSE 1710
1710        IF i MOD 32=0 THEN s=1:x1=1:x2=22 [2623]
:x3=23:iu=-28:GOSUB 1880:GOTO 1730 ELSE 17
20
1720        IF i MOD 16=0 THEN s=2:x1=41:x2=6 [2832]
2:x3=63:iu=-12:GOTO 1730 ELSE 1730
1730 NEXT                                     [350]
1740 CLOSEIN                                  [752]
1750 IF dat$="Kanada.dat" THEN GOSUB 1880:   [11021]
o=50:dat$="Nordam.dat":land$="Nordamerika"
:GOTO 1620 ELSE IF dat$="Nordam.dat" THEN
GOSUB 1880:o=33:dat$="Sudam.dat":land$="Su
edamerika":GOTO 1610 ELSE 1760
1760 LOCATE 26,22:PRINT "Keine weiteren Da   [3276]
ten vorhanden.
1770 LOCATE 1,25:PRINT CHR$(24);"            [4932]
 Ende                          Wochentag
          Neustart              ";CHR$(24);
1780 x=12:x1=12                               [986]
1790 LOCATE x1,23:PRINT null$:LOCATE x,23:   [3720]
PRINT pfeil$
1800 WHILE INKEY$="":WEND                     [1607]
1810 IF INKEY(1)=0 THEN GOTO 1820 ELSE IF    [4415]
INKEY(8)=0 THEN GOTO 1830 ELSE IF INKEY(18
)=0 THEN GOTO 1840 ELSE 1800
1820 IF x=12 THEN x1=x:x=38:GOTO 1790 ELSE   [5387]
 IF x=38 THEN x1=x:x=66:GOTO 1790 ELSE IF
x=66 THEN x1=x:x=66:GOTO 1790
1830 IF x=66 THEN x1=x:x=38:GOTO 1790 ELSE   [6751]
 IF x=38 THEN x1=x:x=12:GOTO 1790 ELSE IF
x=12 THEN x1=x:x=12:GOTO 1790
1840 IF x=12 THEN CALL 0 ELSE IF x=66 THEN   [1871]
 MODE 1:GOTO 270 ELSE 580
1850 SOUND 1,1000,200:IF INKEY(18)=0 THEN    [14534]
CALL 0 ELSE IF INKEY(47)=0 THEN LOCATE INT
((40-(20+LEN(day$)+LEN(a$)))/2),12:PRINT S
TRING$((25+LEN(day$)+LEN(a$)),"  "):TAG:MO
VE 232,272:PRINT STRING$(10," ");:TAGOFF
1860 LOCATE 1,25:PRINT STRING$(40," ");:GO   [2181]
TO 750
1870 END                                      [110]
1880 LOCATE 30,23:PRINT"Bitte Taste drueck   [13641]
en !":CALL &BB18:LOCATE 30,23:PRINT STRING
$(22," "):FOR ii=1 TO 16:LOCATE 1,ii+4:PRI
NT STRING$(19," "):LOCATE 22,ii+4:PRINT"
     ":LOCATE 41,ii+4:PRINT STRING$(19,"
"):LOCATE 62,ii+4:PRINT"          ":NEXT
1890 RETURN                                   [555]
```

```
10 ' Datei Nr.1 --> AMERIKA                   [1587]
20 '                                          [117]
30 CLS:a=12:dat$="Kanada"                     [1540]
40 DIM aa$(a),t(a)                            [744]
50 OPENOUT dat$+".dat"                        [879]
60 FOR i=1 TO a                               [421]
70    READ aa$(i),t(i)                        [1113]
80    WRITE#9,aa$(i),t(i)                     [1749]
90 NEXT                                       [350]
100 CLOSEOUT                                  [902]
110 LOCATE 1,1:PRINT dat$+".DAT wurde erze    [3330]
ugt."
120 IF dat$="Kanada" THEN 130 ELSE IF dat$    [3856]
="Nordam" THEN 140 ELSE 150
130 ERASE aa$,t:dat$="Nordam":a=50:GOTO 40    [2800]
140 ERASE aa$,t:dat$="Sudam":a=33:GOTO 40     [2483]
150 END                                       [110]
160 ' Datas                                   [433]
170 '                                         [117]
180 DATA Newfoundland,-6                      [1528]
190 DATA Nova Scotia,-5                       [1069]
200 DATA New Brunswick,-5                     [1281]
210 DATA Prince Edward Is.,-5                 [1159]
220 DATA Quebec,-6                            [682]
230 DATA Ontario,-6                           [623]
```

Listing TIMER

```
240 DATA Manitoba,-7                          [952]
250 DATA Saskatchewan,-8                      [853]
260 DATA Alberta,-9                           [267]
270 DATA British Columbia,-9                  [883]
280 DATA Yukon Territory,-9                   [2589]
290 DATA Northwest Territory,-8               [857]
300 '                                         [117]
310 ' USA                                     [132]
320 '                                         [117]
330 DATA Alabama,-7                           [349]
340 DATA Alaska,-11                           [411]
350 DATA Arizona,-8                           [93]
360 DATA Arkansas,-7                          [727]
370 DATA Colorado,-8                          [683]
380 DATA Connecticut,-6                       [1650]
390 DATA Delaware,-6                          [767]
400 DATA Florida,-6                           [1328]
410 DATA Georgia,-6                           [902]
420 DATA Hawaii,-11                           [295]
430 DATA Idaho,-9                             [661]
440 DATA Illinois,-6                          [568]
450 DATA Indiana,-6                           [252]
460 DATA Iowa,-7                              [162]
470 DATA Kalifornien,-9                       [243]
480 DATA Kansas,-7                            [412]
490 DATA Kentucky,-6                          [663]
500 DATA Louisiana,-7                         [926]
510 DATA Maine,-6                             [447]
520 DATA Maryland,-6                          [854]
530 DATA Massachusetts,-6                     [674]
540 DATA Michigan,-6                          [652]
550 DATA Minnesota,-7                         [488]
560 DATA Mississippi,-7                       [473]
570 DATA Missouri,-7                          [535]
580 DATA Montana,-8                           [663]
590 DATA Nebraska,-8                          [870]
600 DATA Nevada,-8                            [484]
610 DATA New Hampshire,-6                     [1022]
620 DATA New Jersey,-6                        [516]
630 DATA New York,-6                          [1306]
640 DATA New Mexico,-8                        [1088]
650 DATA North Carolina,-6                    [1057]
660 DATA North Dakota,-7                      [681]
670 DATA Ohio,-6                              [492]
680 DATA Oklahoma,-7                          [105]
690 DATA Oregon,-9                            [356]
700 DATA Pennsylvania,-6                      [1311]
710 DATA Rhode Island,-6                      [1366]
720 DATA South Carolina,-6                    [1537]
730 DATA South Dakota,-7                      [873]
740 DATA Tennessee,-7                         [993]
750 DATA Texas,-7                             [963]
760 DATA Utah,-8                              [684]
770 DATA Vermont,-6                           [912]
780 DATA Virginia,-6                          [1669]
790 DATA Washington,-9                        [1096]
800 DATA West Virginia,-6                     [841]
810 DATA Wisconsin,-6                         [1313]
820 DATA Wyoming,-8                           [732]
830 '                                         [117]
840 ' Sud-,Mittelamerika                      [2230]
850 '                                         [117]
860 DATA Argentinien,-4                       [1640]
870 DATA Bahamas,-6                           [680]
880 DATA Bolivien,-5                          [263]
890 DATA Brasilien,-4                         [471]
900 DATA Chile,-5                             [190]
910 DATA Costa Rica,-7                        [648]
920 DATA Dom. Republik,-6                     [375]
930 DATA Ecuador,-6                           [898]
940 DATA Guatemala,-7                         [780]
950 DATA Guyana,-4                            [254]
960 DATA Haiti,-6                             [71]
970 DATA Honduras,-7                          [884]
980 DATA Jamaica,-6                           [475]
990 DATA Kolumbien,-6                         [700]
1000 DATA Kuba,-6                             [242]
1010 DATA Leeward Islands,-5                  [733]
1020 DATA Mexico,-7                           [855]
1030 DATA Nicaragua,-7                        [380]
1040 DATA Panama,-6                           [502]
1050 DATA Paraguay,-5                         [810]
1060 DATA Peru,-6                             [729]
1070 DATA El Salvador,-6                      [832]
1080 DATA Surinam,-4.5                        [859]
1090 DATA Trinidad,-4                         [803]
1100 DATA Tobago,-4                           [591]
1110 DATA Uruguay,-4                          [665]
```

Listing TIMER

```
1120 DATA Venezuela,-5                          [894]
1130 DATA Windward Islands,-5                   [2038]
1140 DATA Virgin Islands,-5                     [1919]
1150 DATA Cayman Islands,-6                     [423]
1160 DATA Falkland Islands,-5                   [864]
1170 DATA Puerto Rico,-5                        [1208]
1180 DATA Turks & Caicos Is.,-5                 [1762]
```

```
10 ' Datei Nr.2 --> EUROPA                      [1151]
20 '                                            [117]
30 DIM aa$(25),t(25)                            [599]
40 OPENOUT "EUROPA.dat"                         [1254]
50 FOR i=1 TO 25                                [312]
60   READ aa$(i),t(i)                           [1113]
70   WRITE#9,aa$(i),t(i)                        [1749]
80 NEXT                                         [350]
90 CLOSEOUT                                     [902]
100 ' Datas                                     [433]
110 '                                           [117]
120 DATA Albanien,0                             [324]
130 DATA Belgien,0                              [940]
140 DATA Bulgarien,0                            [477]
150 DATA BRD,0                                  [178]
160 DATA Daenemark,0                            [370]
170 DATA DDR,0                                  [398]
180 DATA Finnland,0                             [277]
190 DATA Frankreich,0                           [1012]
200 DATA Griechenland,0                         [1387]
210 DATA Grossbritannien,0                      [1603]
220 DATA Island,0                               [444]
230 DATA Irland,0                               [508]
240 DATA Italien,0                              [290]
250 DATA Jugoslawien,0                          [695]
260 DATA Niederlande,0                          [714]
270 DATA Norwegen,0                             [469]
280 DATA Oesterreich,0                          [813]
290 DATA Polen,0                                [749]
300 DATA Portugal,0                             [1016]
310 DATA Rumaenien,0                            [621]
320 DATA Schweden,0                             [1116]
330 DATA Schweiz,0                              [719]
340 DATA Spanien,0                              [715]
350 DATA CSSR,0                                 [540]
360 DATA Ungarn,0                               [659]
```

```
10 ' Datei Nr.3 --> AFRIKA                      [1311]
20 '                                            [117]
30 DIM aa$(43),t(43)                            [694]
40 OPENOUT "afrika.dat"                         [809]
50 FOR i=1 TO 43                                [534]
60   READ aa$(i),t(i)                           [1113]
70   WRITE#9,aa$(i),t(i)                        [1749]
80 NEXT                                         [350]
90 CLOSEOUT                                     [902]
100 ' Datas                                     [433]
110 '                                           [117]
120 DATA Algerien,0                             [223]
130 DATA Angola,0                               [454]
140 DATA Aethiopien,2                           [316]
150 DATA Botswana,1                             [552]
160 DATA Burundi,2                              [219]
170 DATA Dahomey,2                              [970]
180 DATA Elfenbeinkueste,-1                     [952]
190 DATA Gabun,0                                [342]
200 DATA Gambia,-1                              [79]
210 DATA Ghana,-1                               [228]
220 DATA Guinea,-1                              [669]
230 DATA Kamerun,0                              [181]
240 DATA Kenia,2                                [375]
250 DATA Kongo,0                                [552]
260 DATA Lesotho,1                              [252]
270 DATA Liberia,-0.15                          [997]
280 DATA Libyen,1                               [455]
290 DATA Madagaskar,2                           [1951]
300 DATA Malawi,2                               [858]
310 DATA Mali,-1                                [500]
320 DATA Marokko,-1                             [969]
330 DATA Mauretanien,-1                         [901]
340 DATA Niger,-1                               [766]
350 DATA Nigeria,0                              [880]
360 DATA Obervolta,-1                           [1149]
370 DATA Rhodesien,1                            [1138]
380 DATA Ruanda,2                               [641]
390 DATA Sambia,1                               [755]
400 DATA Senegal,-1                             [361]
```

Listing TIMER

```
410 DATA Somalia,2                              [1006]
420 DATA Sudan,1                                [754]
430 DATA Suedafrika,1                           [948]
440 DATA Tansania,2                             [856]
450 DATA Togo,-1                                [915]
460 DATA Tschad,0                               [715]
470 DATA Tunesien,0                             [957]
480 DATA Uganda,2                               [1108]
490 DATA Aegypten,1                             [430]
500 DATA Komoren,2                              [342]
510 DATA Kapverden,-3                           [483]
520 DATA Mosambik,1                             [540]
530 DATA Kanaren,-1                             [1012]
540 DATA Suedwestafrika,1                       [1459]
```

```
10 ' Datei Nr.4 --> ASIEN                       [952]
20 '                                            [117]
30 DIM aa$(35),t(35)                            [1208]
40 OPENOUT "asien.dat"                          [865]
50 FOR i=1 TO 35                                [322]
60   READ aa$(i),t(i)                           [1113]
70   WRITE#9,aa$(i),t(i)                        [1749]
80 NEXT                                         [350]
90 CLOSEOUT                                     [902]
100 ' Datas                                     [433]
110 '                                           [117]
120 DATA Afghanistan,3.5                        [655]
130 DATA Bahrain,3                              [663]
140 DATA Bhutan,5.5                             [771]
150 DATA Birma,5.5                              [480]
160 DATA Brunei,7                               [1049]
170 DATA Ceylon,4.5                             [1122]
180 DATA China,7                                [103]
190 DATA Indien,4.5                             [335]
200 DATA Indonesien,6                           [1031]
210 DATA Irak,2                                 [153]
220 DATA Iran,2.5                               [238]
230 DATA Israel,1                               [537]
240 DATA Japan,8                                [406]
250 DATA Jemen,2                                [149]
260 DATA Jordanien,1                            [524]
270 DATA Katar,3                                [461]
280 DATA Korea,8                                [426]
290 DATA Kuwait,2                               [636]
300 DATA Laos,7                                 [254]
310 DATA Libanon,1                              [771]
320 DATA Malaysia,7                             [1045]
330 DATA Malediven,4                            [959]
340 DATA Mongolei,7                             [910]
350 DATA Nepal,4.5                              [544]
360 DATA Oman,3                                 [446]
370 DATA Pakistan,4                             [640]
380 DATA Philippinen,7                          [1391]
390 DATA Saudi-Arabien,2                        [986]
400 DATA Sikkim,7                               [702]
410 DATA Syrien,1                               [903]
420 DATA Taiwan,7                               [798]
430 DATA Thailand,6                             [512]
440 DATA Tuerkei,1                              [630]
450 DATA Vietnam,7                              [1043]
460 DATA Zypern,1                               [836]
```

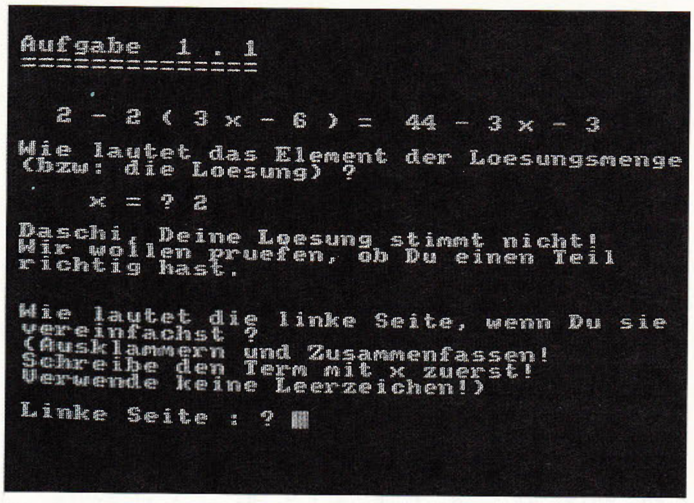
```
10 ' Datei Nr.5 --> AUSTRALIEN                  [851]
20 '                                            [117]
30 DIM aa$(12),t(12)                            [998]
40 OPENOUT "austral.dat"                        [1480]
50 FOR i=1 TO 12                                [321]
60   READ aa$(i),t(i)                           [1113]
70   WRITE#9,aa$(i),t(i)                        [1749]
80 NEXT                                         [350]
90 CLOSEOUT                                     [902]
100 ' Datas                                     [433]
110 '                                           [117]
120 DATA Norfolk Islands,10.5                   [1582]
130 DATA New South Wales,9                      [1583]
140 DATA Queensland,9                           [465]
150 DATA Tasmania,9                             [859]
160 DATA Victoria,9                             [994]
170 DATA New Guinea,9                           [970]
180 DATA Southern Australia,9.5                 [1884]
190 DATA Northern Territory,9.5                 [969]
200 DATA Papua,8                                [815]
210 DATA Western Australia,7                    [830]
220 DATA Christmas Islands,6                    [997]
230 DATA Cocos Islands,5.5                      [746]
```

Listing TIMER

LIGL

Ein Trainingsprogramm für lineare Gleichungen

Mathematik ist wie Fahrradfahren. Ganz verlernt man es zwar nie, doch wenn es nicht ab und zu praktiziert wird, bedarf es schon der einen oder anderen Übung, um wieder "voll dabei zu sein".



Ein Lernprogramm für lineare Gleichungen hilft Ihnen, Gleichungen besser zu verstehen

Besondere Schwierigkeiten in der Mathematik bereitet vielen das Rechnen mit Buchstaben und Unbekannten – mit Zahlen zu rechnen stellt ja seit der Erfindung des Taschenrechners niemanden mehr vor irgendein Problem. Wenn es dann aber um Umformungen, Platzhalter und "häßliche kleine Ixe" geht, dann hilft der Rechner aus der Jackentasche plötzlich nicht mehr weiter, und man muß die eigenen Gehirnzellen ein wenig auf Trapp bringen – Unbelehrbare berechnen zwar heute noch Gleichungen mit x per Taschenrechner, doch wird deren Anzahl stetig geringer, da auch sie irgendwann merken, daß da auf der Tastatur ja kein x, sondern ein Malzeichen ist.

Ein immer wieder gern gesehenes Thema der Mathematik ist das Rechnen mit linearen Gleichungssystemen. Sie sind in hervorragender Weise dazu geeignet, erfrischend auf die mathematisch orientierten Gehirnzellen zu wirken. Das Programm LIGL nimmt nun die Funktion des Lehrers ein, der Ihnen Aufgaben stellt und Fragen beantwortet.

Der Computer gibt nacheinander lineare Gleichungen aus und fragt nach der Lösung. Wird diese nicht richtig eingegeben, so prüft er, ob zumindest die linke und die rechte Seite der ursprünglichen Gleichung richtig vereinfacht wurde (Auflösung von Klammern; Zusammenfassen der Terme) und gibt anderenfalls hierfür das richtige Ergebnis aus.

Falls diese Hilfe noch nicht dazu ausreicht, daß der Lernende die richtige Lösung findet, so gibt der Computer genaue Anweisungen, mit welchen Rechenschritten (Äquivalenzumformungen) die umgeformte Gleichung zu lösen ist.

Folgende Typen von Gleichungen werden ausgegeben:

1) $a(b+cx) = d-(ex-f)$
2) $a-b(cx-d) = e-fx-g$
3) $ax-(b+cx) = (dx+e)f+g$
4) $-(ax+b)+c(d-ex) = x-(f+gx)$
5) $a(b-cx)-(dx+e) = f(gx-7)$

Für die Variablen a,b,c,d,e,f,g werden dabei jeweils ganze Zahlen eingesetzt, und zwar so, daß sich als Lösung (für x) eine ganze Zahl zwischen -20 und 20 ergibt.

Gegenüber vergleichbaren Programmen, die nur Aufgaben generieren und danach die Richtigkeit der Lösung überprüfen, hat das Programm LIGL zwei Besonderheiten:

- Durch geschickte Wahl der Zufallszahlen wird erreicht, daß alle Rechnungen innerhalb der ganzen Zahlen ausgeführt werden können; insbesondere sind alle Lösungen ganzzahlig.
- Wenn die Lösung nicht sofort gefunden wurde, stellt das Programm ein gestuftes System von Hilfen bereit. Der Lernende kann also auch positive Rückmeldung erhalten, wenn er nur Teile der Aufgabe richtig löst.

(Dr. Walter Witzel/jf/cd)

für 464-664-6128

```
10 REM TRAININGSPROGRAMM  LINEARE GLEICHUN [3580]
GEN ...........................Nov.88
20 REM .................................... [4158]
..................Autor.Walter Witzel
22 MODE 1                                   [506]
25 LOCATE 10,4:PRINT"Trainingsprogramm"     [2983]
27 LOCATE 9,6:PRINT"LINEARE GLEICHUNGEN"    [2823]
30 FOR typ=1 TO 5:READ n(typ):NEXT typ      [3471]
35 DATA 450,550,650,750,850                 [533]
37 LOCATE 2,9:PRINT"Nimm Zettel und Bleist  [4936]
ift zur Hand und"
38 PRINT"gib danach Deinen Namen ein (mit   [3698]
RETURN)"
40 INPUT n$                                 [497]
70 CLS:PRINT"Vorbemerkungen:":PRINT"======  [3734]
=========":PRINT
80 PRINT"Der Computer gibt Dir im folgende  [3119]
n"
85 PRINT"nacheinander lineare Gleichungen   [4478]
aus."
90 PRINT"Die Variable ist dabei jeweils  x  [6492]
 ;     Grundmenge sind die rationalen Zahl
en."
```

Listing LIGL

```
100 PRINT:PRINT"Du sollst diese Aufgaben l  [3210]
oesen"
105 PRINT"(die Loesungsmenge bestimmen) un  [6447]
d das   Ergebnis eingeben."
110 PRINT:PRINT"Sollte Deine Loesung einma  [6246]
l nicht stim- men, so erhaelst Du vom ";
120 PRINT"Computer Hilfen."                 [2782]
180 GOSUB 990                               [1014]
200 REM Wahl des Schwierigkeitsgrads  ****  [3944]
*************************************
210 MODE 1:PRINT n$;", willst Du mit":PRIN  [3278]
T
212 PRINT"1   leichteren Aufgaben"          [2426]
214 PRINT"2   schwierigeren Aufgaben"       [3271]
216 PRINT:PRINT"beginnen ? (1 oder 2 einge  [3687]
ben)"
217 a$=INKEY$:IF a$<>"1" AND a$<>"2" THEN   [1214]
217
218 l=VAL(a$)                               [795]
250 REM Beginn eines neuen Aufgabentyps  *  [4084]
************************************
260 i=0:r=0:rr=0:ff=0                       [586]
300 REM Neue Aufgabe ********************   [3743]
************************************
310 CLS                                     [91]
320 i=i+1                                   [444]
330 PRINT"Aufgabe ";l;".";i                 [2001]
335 PRINT"==============":PRINT             [1635]
```

Listing LIGL

```
340 x=INT(RND*40)-19                          [1631]
350 a= INT(8*RND)+2                           [1298]
352 b= INT(8*RND)+2                           [1056]
354 c= INT(8*RND)+2                           [1342]
356 d= INT(19*RND)-9:IF ABS(d)<2 THEN 356     [3283]
362 e= INT(19*RND)-9:IF ABS(e)<2 THEN 362     [2203]
364 f= INT(19*RND)-9:IF ABS(f)<2 THEN 364     [1489]
380 IF l=1 THEN z=(i MOD 3)+1:ON z GOTO 40    [2895]
0,500,600
390 IF l=2 THEN z=(i MOD 2)+1:ON z GOTO 70    [2683]
0,800
400 REM Aufgabentyp 1 ******************* [3533]
***************************************
405 typ=1                                     [537]
410 IF a*d+f =0 THEN f=f+1                    [916]
420 g=x*(a*d+f)-b+d*e                         [1323]
430 r1=-f :r2=g+b                             [394]
435 l1=a*d:l2=d*e                             [1036]
440 GOSUB 450:GOTO 1000                       [1978]
450 LOCATE 2,5                                [635]
460 PRINT d;"(";e;"+";a;"x ) =";g;" - (";f   [3080]
;"x -";b;")"
480 RETURN                                    [555]
500 REM Aufgabentyp 2 ******************* [3371]
***************************************
505 typ =2                                    [544]
510 IF c-b*e=0 THEN c=c-1                     [268]
520 g=f+b*a+d+x*(c-b*e)                       [1680]
530 l1=-b*e :l2=f+b*a                         [743]
535 r1= -c  :r2=g-d                           [517]
540 GOSUB 550:GOTO 1000                       [1058]
550 LOCATE 2,5                                [635]
560 PRINT f;"-";b;"(";e;"x -";a;") = ";g;"   [2104]
-";c;
570 IF d>0 THEN PRINT"x -";d ELSE PRINT"x    [2229]
+";-d
580 RETURN                                    [555]
600 REM Aufgabentyp 3 ******************* [3949]
***************************************
605 typ=3                                     [543]
610 IF d-a-f*b=0 THEN a=a+1                   [1171]
620 g=e+c*b-x*(d-a-f*b)                       [1238]
630 l1=d-a:   l2=-e                           [748]
635 r1=f*b:   r2=c*b-g                        [1028]
640 GOSUB 650:GOTO 1000                       [1140]
650 PRINT:PRINT:PRINT                         [1082]
660 PRINT d;"x -(";e;"+";a;"x) = (";f;"x +   [2785]
";c;")";b;
670 IF g>0 THEN PRINT"-";g ELSE PRINT "+";    [2072]
-g
680 RETURN                                    [555]
700 REM Aufgabentyp 4 ******************* [3492]
***************************************
705 typ=4                                     [534]
710 IF c-d-e*b-1=0 THEN c=c+1                 [1670]
720 g=a-e*f-x*(c-1-d-b*e)                     [1500]
730 l1=-d-e*b:l2=e*f-a                        [2071]
735 r1=1-c    :r2=-g                          [1312]
740 GOSUB 750:GOTO 1000                       [1206]
750 PRINT:PRINT:PRINT                         [1082]
755 u=ABS(d):GOSUB 885                        [1312]
758 IF d>0 THEN PRINT "-(";z$; ELSE PRINT     [1684]
"-(-";z$;
760 PRINT "x +";a;")";                        [1011]
762 u=ABS(e):GOSUB 885                        [788]
765 IF e>0 THEN PRINT "+";z$; ELSE PRINT "    [2388]
-";z$;
768 u=ABS(c):GOSUB 885                        [1148]
770 PRINT"(";f;"-";b;"x) = x -(";g;"+";z$;    [2107]
"x)"
780 RETURN                                    [555]
800 REM Aufgabentyp 5 ******************* [4013]
***************************************
805 typ=5                                     [533]
810 IF e*c+d+a*f=0 THEN d=d+1                 [592]
820 g=x*(e*c+d+a*f)-7*a-e*b                   [1383]
830 l1=-e*c-d:l2=g+e*b                        [1752]
835 r1=a*f    :r2=-7*a                        [1210]
840 GOSUB 850:GOTO 1000                       [1684]
850 PRINT:PRINT:PRINT                         [1082]
855 u=ABS(e):GOSUB 885:IF e<0 THEN z$="-"+    [2246]
z$
858 z$=z$+"("                                 [331]
860 PRINT z$;b;"-";                           [558]
862 u=ABS(c):GOSUB 885:PRINT z$;"x) - (";d    [2691]
;
865 u=ABS(g):GOSUB 885                        [668]
```

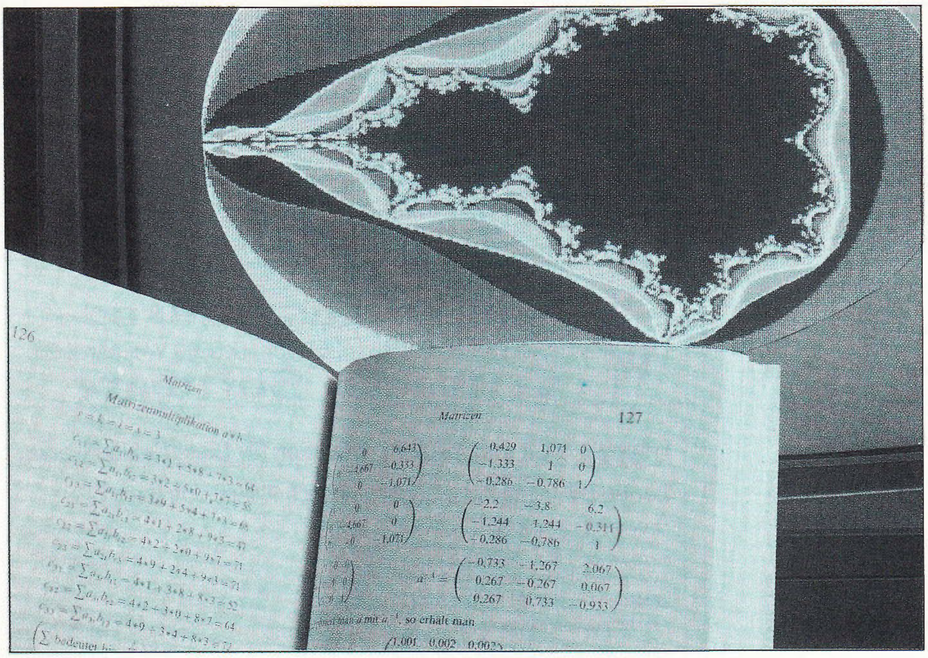
Listing LIGL

```
870 IF g<0 THEN PRINT"x +";z$; ELSE PRINT"    [3467]
x -";z$;
875 PRINT") =";a;"(";f;"x";" -7)"             [813]
880 RETURN                                    [555]
885 REM Zahl u in Zeichenkette z$ verwande    [3671]
ln **********************************
886 t=INT(u/1000):h=INT(u/100)-10*t:z=INT(    [1877]
u/10)-100*t-10*h
888 E1=INT(u)-1000*t-100*h-10*z               [1529]
890 z$=""                                     [449]
892 IF t>0 THEN z$=z$+CHR$(48+t)              [642]
893 IF h>0 OR t>0 THEN z$=z$+CHR$(48+h)       [1444]
894 IF z>0 OR h>0 OR t>0 THEN z$=z$+CHR$(4    [2621]
8+z)
895 IF E1>0 OR z>0 OR h>0 OR t>0 THEN z$=z    [2683]
$+CHR$(48+E1)
898 RETURN                                    [555]
900 REM Linke und rechte Seite der vereinf    [5578]
achten Gleichung ******************
905 l$="":r$="":l1$="":r1$="":l10=l1:l20=l    [4202]
2:r10=r1:r20=r2
910 IF l1<0 THEN l$="-":l1=-l1                [1221]
920 u=l1:GOSUB 885:IF l1<>0 THEN l$=l$+z$+    [2168]
"x"
930 IF l2<0 THEN l$=l$+"-":l1$="-":l2=-l2     [1692]
933 IF l20>0 THEN l$=l$+"+":l1$="+"           [1943]
935 u=l2:GOSUB 885:IF l2<>0 THEN l$=l$+z$:    [2727]
l1$=l1$+z$
950 IF r1<0 THEN r$="-":r1=-r1                [1796]
955 u=r1:GOSUB 885:IF r1<>0 THEN r$=r$+z$+    [3098]
"x"
960 IF r20<0 THEN r$=r$+"-":r1$=r1$+"-":r2    [1078]
=-r2
963 IF r20>0 THEN r$=r$+"+":r1$="+"           [1442]
965 u=r2:GOSUB 885:IF r2<>0 THEN r$=r$+z$:    [2131]
r1$=r1$+z$
970 IF l10=1 THEN l1$="x"+l1$                 [861]
972 IF l10=-1 THEN l1$="-x"+l1$               [939]
974 IF r10=1  THEN r1$="x"+r1$                [1391]
976 IF r10=-1 THEN r1$="-x"+r1$               [589]
980 RETURN                                    [555]
990 REM UPR PAUSE ********************** [3317]
***************************************
993 PRINT:PRINT:PRINT:PRINT"--- Bitte beli    [5527]
ebige Taste druecken! ---"
995 a$=INKEY$:IF a$="" THEN 995               [1101]
998 RETURN                                    [555]
1000 REM Loesung annehmen und ueberpruefen    [3743]
 **********************************
1005 lg=l1*x+l2:rg=r1*x+r2:GOSUB 900          [1083]
1010 PRINT:PRINT"Wie lautet das Element de    [5948]
r Loesungsmenge(bzw: die Loesung) ?"
1020 PRINT:INPUT "      x = ";xs              [621]
1030 IF xs=x THEN 1200                        [1760]
1040 PRINT:PRINT n$;", Deine Loesung stimm    [3732]
t nicht!":rr=0:ff=ff+1
1045 PRINT"Wir wollen pruefen, ob Du einen    [5253]
 Teil     richtig hast.":PRINT
1050 PRINT:PRINT"Wie lautet die linke Seit    [5653]
e, wenn Du sie vereinfachst ?
1052 PRINT"(Ausklammern und Zusammenfassen    [4559]
!"
1054 PRINT"Schreibe den Term mit x zuerst!    [6847]
         Verwende keine Leerzeichen!)"
1060 PRINT:INPUT "Linke Seite : ";ls$         [1593]
1065 IF l$=ls$ OR l1$=ls$ THEN PRINT"Richt    [3361]
ig,";n$ :GOTO 1070
1068 PRINT "Die linke Seite":PRINT"lautet     [3296]
vereinfacht: ";l$
1070 PRINT:PRINT"Wie lautet entsprechend v    [5569]
ereinfacht die rechte Seite ?":PRINT
1075 INPUT "Rechte Seite : ";rs$              [1684]
1080 IF rs$=r$ OR r1$=rs$ THEN PRINT"Stimm    [1616]
t,";n$:GOTO 1088
1084 PRINT "Die rechte Seite":PRINT"lautet    [4921]
 vereinfacht: ";r$
1088 GOSUB 990                                [1014]
1090 CLS:PRINT "Die Gleichung ":LOCATE 1,1    [4687]
:ON typ GOSUB 450,550,650,750,850
1095 PRINT:PRINT"laesst sich also auf folg    [5258]
ende Form      bringen":PRINT
1100 PRINT"     ";l$;"   =   ";r$:PRINT       [1656]
1110 PRINT"Wie lautet die Loesung ?"          [3440]
1120 PRINT:PRINT:INPUT "   x = ";xs           [1436]
1130 IF xs=x THEN PRINT n$;",jetzt stimmt     [4099]
es.":GOSUB 990:GOTO 1300
1135 IF xs=-x THEN 1192                       [1078]
```

Listing LIGL

```
1140 PRINT"Das stimmt nicht.":PRINT            [1999]
1142 PRINT"Forme die Gleichung wie folgt u [1632]
m:":PRINT
1144 IF r10<0 THEN PRINT"1) Addiere auf be [4456]
iden Seiten  ";-r10;"x !"
1146 IF r10>0 THEN PRINT"1) Subtrahiere au [5299]
f beiden Seiten ";r10;"x !"
1148 IF 120<0 THEN PRINT"2) Addiere auf be [5685]
iden Seiten ";-120
1150 IF 120>0 THEN PRINT"2) Subtrahiere au [7064]
f beiden Seiten ";120
1152 PRINT"3) Teile beide Seiten durch ";1 [4431]
10-r10
1154 PRINT:PRINT"Wie lautet die Loesung?": [4093]
PRINT
1160 INPUT "    x = ";xs                   [1192]
1170 IF x=xs THEN PRINT "Richtig!!!":GOSUB [3168]
 990:GOTO 1300
1175 IF xs=-x THEN 1192                    [1078]
1180 PRINT"Richtig ist:   x =";x           [1698]
1190 GOSUB 990 : GOTO 1300                 [1145]
1192 PRINT"Du hast den Betrag richtig, abe [5540]
r das     Vorzeichen stimmt nicht "
1194 PRINT"Richtig ist    x =";x           [1936]
1196 GOSUB 990:GOTO 1300                   [1145]
1200 REM Rueckmeldung bei richtiger Loesun [4321]
g ***********************************
1210 ff=0:rr=rr+1:r=r+1                    [844]
1220 z=INT(5*RND)+1: ON z GOTO 1230,1240,1 [2390]
250,1260,1270
1230 PRINT"Richtig gerechnet,";n$:GOTO 128 [3969]
0
1240 PRINT"Das stimmt so."        :GOTO 128 [2932]
0
1250 PRINT"Bravo,";n$;", das hast Du gut g [4105]
emacht":GOTO 1280
1260 PRINT"Stimmt! Wieder ein Pluspunkt fu [3735]
er Dich!":GOTO 1280
1270 PRINT"Richtig. Weiter so, ";n$        [3093]
1280 GOSUB 990: GOTO 1300                  [1145]
1300 REM                                   [272]
1400 REM Wahl des naechsten Aufgabentyps * [3755]
*************************************
1410 IF i<10 AND rr<3 AND ff<3 THEN 300:RE [3397]
M naechste Aufgabe von gleichen Typ
1415 IF i>9 THEN 1500                      [1191]
1420 IF rr=3 AND l=1 THEN l=2:PRINT"Es fol [6086]
gen etwas schwierigere Aufgaben":GOTO 250
1430 IF ff=3 AND l=2 THEN l=1:PRINT"Es fol [5703]
gen etwas einfachere Aufgaben":GOTO 250
1440 IF rr>=5 THEN PRINT n$;", Du bist fue [5571]
r dies Programm zu gut":GOTO 1500
1460 GOTO 300                              [504]
1470 PRINT"Du hast jetzt mehrere Aufgaben  [6244]
nachein- ander falsch geloest."
1480 PRINT "Willst Du trotzdem weitermache [4614]
n? (j/n)";
1485 a$=INKEY$:IF a$="" THEN 1485          [1432]
1490 IF a$="j" OR a$="J" THEN 300          [1343]
1495 IF a$="n" OR a$="N" THEN 1500         [1642]
1498 GOTO 1480                             [307]
1500 REM Ergebnis ausgeben *************** [4040]
*************************************
1510 GOSUB 990:MODE 1                      [1072]
1520 PRINT"Du hast bei den ";              [1345]
1525 IF l=1 THEN PRINT"leichteren " ELSE P [5070]
RINT"schwierigeren "
1530 PRINT "Aufgaben":PRINT"insgesamt ";i; [4193]
" bearbeitet."
1550 PRINT"Davon hast Du ";r;" auf Anhieb  [6676]
richtig    geloest; das sind ";
1560 PRINT INT(1000*r/i)/10;" Prozent.":PR [3012]
INT:PRINT
1570 PRINT "Willst Du weiter mit diesem Pr [5063]
ogramm     arbeiten? (j/n)";
1580 a$=INKEY$:IF a$="" THEN 1580          [1429]
1590 IF a$="j" OR a$="J" THEN 200          [1347]
1592 IF a$="n" OR a$="N" THEN END          [1541]
1600 GOTO 1570                             [385]
2050 PRINT"Wie lautet die linke Seite, wen [9828]
n Du sie vereinfacht hast (Ausklammern ";
2052 PRINT"und":PRINT"Zusammenfassen)":PRI [2929]
NT
2054 PRINT"(Schreibe den Term mit x zuerst [5894]
!        Verwende keine Leerzeichen!)"
```

Listing LIGL



Wunderbare Welt der Mathematik

APFELMANN oder In die Tiefe geschaut

Man sollte es nicht für möglich halten, welchen Effekt ein paar, auf den ersten Blick unverständliche, mathematische Formeln auf dem Bildschirm erzeugen, wenn sie sich durch Grafik darstellen dürfen. Heraus kommen sogenannte fraktale Grafiken, besser bekannt als 'Apfelmännchen'.

Geben Sie Listing 1 (APFELMAN.BAS) ein und speichern Sie es unter gleichem Namen. Listing 2 erzeugt die Datei APFELMAN.BIN, Listing 3 die Datei APFELMAN.DAT (dazu später mehr). Haben Sie nun alle drei Programme abgespeichert (bei Kassettenbenutzern bitte in der obigen Reihenfolge hintereinander), starten Sie das Programm APFELMAN.BAS. Dieses lädt selbständig die Programmteile APFELMAN.BIN und APFELMAN.DAT in den Speicher. APFELMAN.BIN enthält die RSX-Befehlserweiterungen (|BUSY und |HARDCOPY), welche folgende Funktionen haben: |BUSY testet, ob der Drucker bereit ist, Daten zu empfangen. Der Befehl |HARDCOPY druckt das Bild im aktuellen Modus aus. Die Datei APFELMAN.DAT muß dabei im Speicher stehen und zwar ab Adresse 8000 hex. In ihr sind die Druckervariablen und die Steuerzeichen für die Escape-Sequenzen (genaueres hierzu unter Drucker: Anpassung) enthalten.

Haben Sie das Programm nun gestartet, schaltet der Computer zunächst auf den zweiten Bildschirmspeicher (CALL &BC06,&40) um. Der Poke-Befehl in Zeile 180 verhindert, daß der MODE-Befehl das aktuelle Bild löscht. Der Bildschirm ist in drei Fenster unterteilt. Im Fenster Nr. 1 stehen immer Informationen zum Programm. Fenster Nr. 2 (Statusfenster) enthält die Werte von Xmin, Xmax, Ymin, Ymax und der Tiefe Tmax. Diese Werte geben die Grenzen des Ausschnittes an, der als 'Apfelmännchen'-Bild dargestellt werden soll.

ESC gibt an, ob eine Unterbrechung während des Zeichnens und Berechnens der Grafik zugelassen wird. JA heißt, daß unterbrochen werden kann, bei NEIN ist es logischerweise nicht möglich, wobei hier darauf zu achten ist, das auch der Dreifingergriff (CRTL-SHIFT-ESC) nicht funktioniert.

MODUS gibt an, in welchem Bildschirmmodus das Bild gezeichnet werden soll. Das Programm unterstützt alle drei Modi. Eine Besonderheit des Programms ist, daß die Farben im Modus 0 entweder an einen Grünmonitor oder an einen Farbmonitor angepaßt sind. Im Fenster 0 stehen die Menü- bzw. Submenüpunkte, die Sie mit den Cursortasten und Copy (nicht Enter) oder dem Joystick anwählen können.

Falls Sie versehentlich den falschen Menüpunkt angewählt haben, gehen Sie im Submenü nur auf Hauptmenü, und schon gelangen Sie in eben dieses zurück.

Doch nun gehen wir näher auf die Bedienung des Programms ein und erklären Ihnen die einzelnen Menüpunkte.

Menüpunkt Bild laden: Sie können ein abgespeichertes Bild von Kassette/Diskette laden. Geben Sie den Dateinamen ohne Anhang (Extension) an, und drücken Sie ENTER. Drücken Sie bei der Abfrage nach dem Namen nur ENTER, so bricht der Computer ab und kehrt ins Submenü zurück.

Bei korrekter Eingabe lädt der CPC die Dateien NAME.DAT und NAME.PIC. NAME.DAT enthält folgende Variablen: xmin!, xmax!, ymin!, ymax, tmax%, dx, dy, cx, cy, modus, zeile%, spalte%, endebild, xleft!, timax%, monitor$, esc$. xmin!, xmax!, ymin!. 'ymax!' und 'tmax%' sind die Grenzen des Bildes mit der Tiefe 'tmax%'. 'cx' und 'cy' sind beim Zeichnen der Grafik die Koordinaten der Bildpunkte im zu berechnenden Ausschnitt (bei erstmaligem Start ist cx=xmin! und cy=ymin!). 'dx' berechnet sich aus folgender Formel: (xmax!-xmin!)/Anzahl der Bildpunkte in x-Richtung im aktuellen Modus-1. 'dy' berechnet sich folgendermaßen: (ymax!-ymin!)/Anzahl der Bildpunkte in y-Richtung-1 (199). 'dx' und 'dy' geben an, um wieviel 'cx' bzw. 'cy' erhöht werden, wenn ein neuer Bildpunkt berechnet wird. 'modus' gibt den Zeichenmodus an, in dem das Bild berechnet werden soll (Standard: 1). 'zeile%' und 'spalte%' sind die Koordinaten des gerade berechneten Bildpunktes auf dem Bildschirm. 'endebild' gibt dabei an, ob das Bild schon fertig gezeichnet ist (Wert: 1) oder ob beim Zeichnen unterbrochen wurde. Solange ein Bild noch nicht fertiggezeichnet ist, bleiben diese Werte immer gleich. Dies hat den Vorteil, daß die Grenzen eines Ausschnittes neu eingegeben werden können, der Computer das Bild aber immer nach den alten Koordinaten berechnet. 'monitor$' gibt an, mit welchem Monitor (Farbe oder Grün) gearbeitet wird. Diese Variable hat nur Auswirkungen beim Darstellen bzw. Zeichnen eines Bildes in Modus 0. Die Farben werden dann entweder für einen Grün- oder für einen Farbmonitor angepaßt. 'esc$' sagt dem Computer, ob beim Zeichnen eine Unterbrechung zugelassen werden soll. Die Datei NAME.PIC enthält das eigentliche Bild. Sie benötigt 17 kByte Speicherplatz auf der Diskette, auf Kassette ist sie acht Blöcke lang.

Speichern: Wenn ein Bild vorhanden ist, speichert der Computer die für die Grafik benötigten Variablen (siehe oben) in der Datei NAME.DAT und das eigentliche Bild in der Datei NAME.PIC ab.

Löschen: Wenn Ihnen ein Ausschnitt aus der 'Apfelmännchen'-Menge nicht gefällt, können Sie das Bild einfach löschen. Sie müssen dann die Sicherheitsabfrage nur mit "J" für "JA" beantworten, bei Eingabe anderer Zeichen kehrt der Computer zum Submenü zurück.

Ausschnitt vergrößern: Ein Ausschnitt kann nur vergrößert werden, wenn ein Bild schon im Speicher vorhanden ist. (Variable 'bild' muß dabei auf "1" stehen) Der Computer schaltet dann auf den ersten Bildschirmspeicher um (in ihm sehen wir unser Apfelmännchen). Sie können nun mittels der Cursortasten oder des Joysticks einen Ausschnitt bestimmen, den Sie vergrößern wollen. Hierzu bewegen Sie den auf dem Schirm angezeigten Cursor an die Stelle, welche die obere linke Ecke werden soll. Betätigen Sie nun die COPY-Taste (bei Joystick den Feuerknopf), und verfahren Sie genauso für die untere rechte Ecke des Ausschnitts. Nachdem Sie so den gewünschten Ausschnitt bestimmt haben, können Sie ihn durch Anwahl des Menüpunktes **Apfelmännchen zeichnen** berechnen und zeichnen lassen.

Apfelmännchen zeichnen: Der Computer berechnet ein Bild aus den Werten 'xmin!', 'xmax!', 'ymin!', 'ymax!' und 'tmax%', falls 'endebild' auf "1" gesetzt ist. Ansonsten fährt der Computer beim Berechnen eines Bildes an der Stelle fort, an der er zuletzt aufgehört hat.

Menüpunkt: Drucker

Anpassung: Die Hardcopyroutine des Programms kann an Ihren Drucker angepaßt werden. Sie ist für dem Epson-kompatiblen DMP-2000 geschrieben. Der Computer muß dabei die Länge folgender Druckersequenzen und Ihre Steuerzeichen von Ihnen wissen:

a) 17/216 Zoll Zeilenabstand
b) 1/216 Zoll Zeilenabstand
c) Grafik in doppelter Dichte (max. 960 Punkte/Zeile) mit 800 Punkten/Zeile
d) 1/8 Zoll Zeilenabstand
e) Seitenvorschub
f) Wagenrücklauf und Zeilenvorschub
g) Druckerreset

Die Länge der Sequenzen ist beliebig (maximal jedoch 255 Zeichen). Sie dürfen jedoch das Zeichen mit der Nummer 255 nicht enthalten, da dieses das Ende der Sequenz der Hardcopyroutine mitteilt.

Hardcopy: Dieser Submenüpunkt kann nur dann aufgerufen werden, wenn der Drucker bereit ist, Daten zu empfangen. Wenn das Bild noch nicht fertiggezeichnet wurde, fragt das Programm nach, ob das Bild auch wirklich ausgedruckt werden soll. Der Computer schaltet dann auf den ersten Bildschirmspeicher um, stellt den Modus ein und setzt die Farben auf Ihre dem Modus angepaßten Werte. Der CPC gibt dann das Bild als DIN-A4-Hardcopy auf dem Drucker aus. Die Hardcopy-Routine kann jederzeit unterbrochen werden. Unter den Ausdruck werden die Grenzen des Ausschnittes und die Tiefe gedruckt, ein Seite vorgeschoben und der Drucker zurückgesetzt.

Druckerparameter speichern: Die Druckervariablen aus dem Submenüpunkt Anpassung werden unter dem Namen APFELMAN.DAT abgespeichert. Sie werden ebenfalls bei erneutem Start dieses Programms automatisch nachgeladen, so daß eine Druckeranpassung nicht jedesmal erfolgen muß.

Menüpunkt: Programm

Ende: Möchten Sie das Programm beenden, beantworten Sie die Sicherheitsabfrage mit "J". Der erste Bildschirmspeicher mit der Grafik wird eingeschaltet und das Programm beendet. Geben Sie bei der Abfrage andere Zeichen ein, so fährt das Programm beim Submenü fort.

Neustart: Das Programm wird neu gestartet. Die Druckervariablen werden neu eingelesen, und der erste Bildschirmspeicher wird gelöscht. Vor dem Neustart sollten Sie eine eventuell vorhandene Grafik abspeichern, da diese ansonsten verlorengeht. Die Variablen 'xmin!', 'xmax!', 'ymin!', 'ymax!' und 'tmax%' werden auf ihre ursprünglichen Werte gebracht. Die Werte sind die Grenzen der Menge, die dargestellt werden soll. Sie können aber Werte verwenden, die außerhalb dieser Menge liegen, nur wird das Apfelmännchen dann nicht unbedingt schön aussehen.

Menüpunkt: Sonstiges

Umschalten Farb-/Grünmonitor: Dieser Submenüpunkt macht sich nur in Mode 0 bemerkbar. Die Farben werden an einen Farb- bzw. Grünmonitor angepaßt. Bei einem Grünmonitor werden die Farben mit zunehmender Farbnummer immer heller, was für einen relativ stufenlosen Farbübergang von dunkel nach hell gedacht ist.

ESC ein/aus: Soll beim Zeichnen eines Apfelmännchens eine Unterbrechung zugelassen werden, steht im Statuswindow bei ESC: 'JA', das bedeutet, daß Sie das Zeichnen eines Bildes unterbrechen und zu anderer Zeit weitermachen können. Dieser Menüpunkt hat auf andere Menüpunkte keine Auswirkung.

Darstellungsmodus: Sie können mit den Cursortasten und Copy bzw. dem Joystick den Modus anwählen, in dem die Grafik dargestellt werden soll. Der Darstellungsmodus wird im Statuswindow unter MODUS angezeigt.

Werte verändern: Dies ist einer der wichtigsten Submenüpunkte des Programms. Sie können die Werte von 'xmin!', 'xmax!', 'ymin!', 'ymax!' und

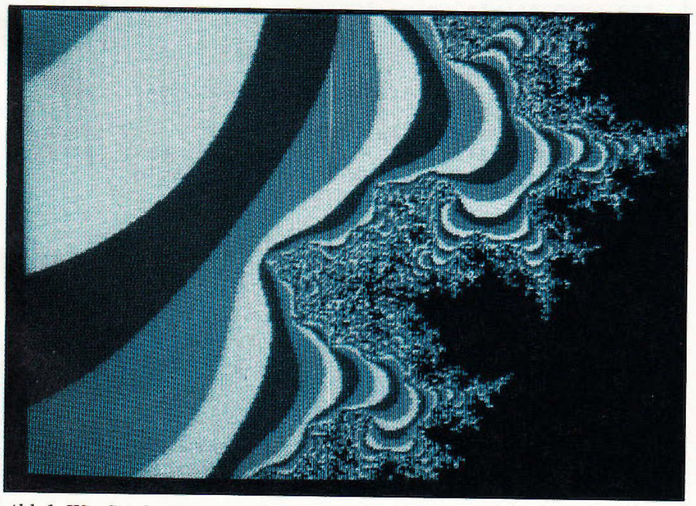


Abb.1: Wie Sie die Werte verändern, bleibt Ihnen überlassen, das Ergebnis...

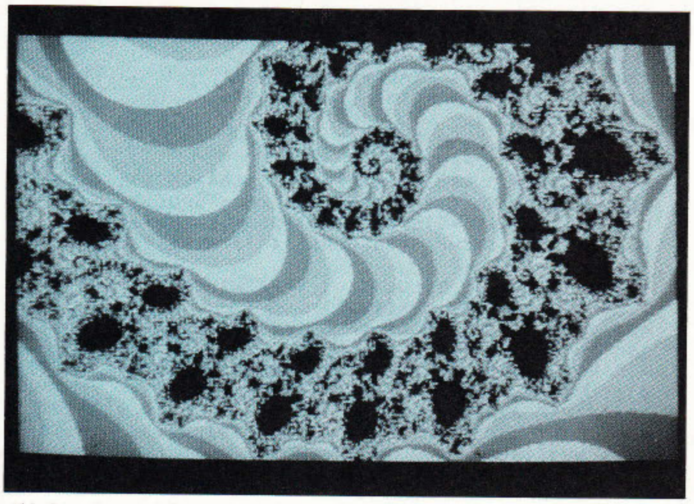


Abb.2: ...kann sich in jedem Falle sehen lassen

'tmax%' einzeln verändern, indem Sie die zu ändernde Variable mit den Cursortasten oder dem Joystick anwählen. Haben Sie alle Werte nach Ihrem Belieben verändert, drücken Sie ENTER. Der CPC fragt Sie dann, ob die Werte noch berichtigt werden sollen, um verzerrtes Zeichnen eines Bildes zu verhindern.

Bild anzeigen: Das Apfelmännchenbild wird im aktuellen Modus angezeigt. Drücken Sie eine Taste, kehrt das Programm zum Submenü zurück.

Soweit also unsere Führung durch unser Apfelmännchenprogramm. Wir wünschen Ihnen jetzt viel Spaß bei Ihrer Reise durch die wunderbare Welt der Fraktale.

(Thorsten Heit/jb)

für 464-664-6128

```
10 '*****************             [659]
20 '*               *             [175]
30 '* Apfelmaennchen *            [1186]
40 '*   (c) 1988 by  *            [908]
50 '*  Thorsten Heit *            [353]
60 '*               *             [175]
70 '*****************             [659]
80 '                              [117]
90 'Initialisierung               [1300]
100 SYMBOL 255,254,252,248,252,238,199,3,1 [10980]
:MEMORY &9FFF:OPENOUT"x":MEMORY &3FFF:CLOS
EOUT 'Platz fuer 2. Bildschirm reservieren
 und Pfeilzeichen definieren
110 INK 0,0:BORDER 0:CALL &BC06,&C0:MODE 1 [4430]
:CLS:CALL &BC06,&40:CLS:INK 1,14:INK 2,20:
INK 3,26
120 IF PEEK(&A000)=1 THEN 130 ELSE LOAD"!a [9848]
pfelman.bin",&A000:CALL &A000 'RSX-Erweite
rung (BUSY,HARDCOPY) laden und initialisie
ren
130 zeile%=0:spalte%=0:monitor$="Gruen":es [9453]
c$="Ja":xmin!=-2.2:xmax!=1:ymin!=-1:ymax!=
1:tmax%=30:modus=1:endebild=1:busy%=0
140 DEF FNxmin=xmax!-(ymax!-ymin!)*1.6:DEF [4014]
 FNymin=ymax!-(xmax!-xmin!)/1.6
150 DEF FNxmax=xmin!+(ymax!-ymin!)*1.6:DEF [4162]
 FNymax=ymin!+(xmax!-xmin!)/1.6
160 DEF FNzeile=2^modus*160-1:DEF FNschrit [3229]
t=5-2^modus+(modus=1)
170 DEF FNdx=(xmax!-xmin!)/FNzeile:DEF FNd [5249]
y=(ymax!-ymin!)/199:dx=FNdx:dy=FNdy
180 a=PEEK(&BDEC)+256*PEEK(&BDED)+23:POKE  [5164]
&BDEC,a AND 255:POKE &BDED,INT(a/256)
190 ON BREAK GOSUB 2370:KEY DEF 66,0,0,0,0 [1212]
200 LOAD"!apfelman.dat",&8000:GOSUB 2440 ' [3947]
Einlesen Druckervariablen
210 DIM menue$(3,6),modus$(2),wert$(4)     [2484]
220 'Einlesen Menue- und Submenuepunkte    [4135]
230 RESTORE 270                            [797]
240 FOR a=0 TO 3:READ menue$(a,0):FOR b=1  [3332]
TO 6
250 READ menue$(a,b)                       [1038]
260 NEXT b,a                               [372]
270 DATA Bild,Laden,Speichern,Loeschen,Aus [6710]
schnitt vergroessern,Apfelmaennchen zeichn
en,Hauptmenue
280 DATA Drucker,Anpassung,Hardcopy,Drucke [3363]
rparameter speichern,Hauptmenue,"",""
290 DATA Programm,Ende,Neustart,Hauptmenue [2865]
,"","",""
300 DATA Sonstiges,Umschalten Farb-/Gruenm [9320]
onitor,Esc ein/aus,Darstellungsmodus,Werte
 veraendern,Bild anzeigen,Hauptmenue
310 FOR a=0 TO 2:READ modus$(a):NEXT       [2107]
320 DATA "0: 160*200 Punkte, 16 Farben","1 [4460]
: 320*200 Punkte, 4 Farben","2: 640*200 Pu
nkte, 2 Farben"
330 FOR a=0 TO 4:READ wert$(a):NEXT        [2053]
340 DATA Xmin,Xmax,Ymin,Ymax,Tiefe         [1776]
350 'Bildaufbau                            [936]
360 WINDOW#1,1,40,1,4                      [1368]
370 PAPER#1,3:PEN#1,1                      [1901]
380 PRINT#1,CHR$(150)STRING$(38,154)CHR$(1 [11239]
56)CHR$(149)SPC(12)"Apfelmaennchen"SPC(12)
CHR$(149)CHR$(149)SPC(7)"(c)1988 by Thorst
en Heit"SPC(7)CHR$(149)CHR$(147)STRING$(38
,154)CHR$(153);
390 PAPER 2:PEN 1:LOCATE 1,5:PRINT CHR$(15 [3006]
0)STRING$(38,154)CHR$(156)
400 LOCATE 1,19:PRINT CHR$(147)STRING$(38, [2344]
154)CHR$(153)
410 FOR a=6 TO 18:LOCATE 1,a:PRINT CHR$(14 [5078]
9);:LOCATE 40,a:PRINT CHR$(149);:NEXT
420 PAPER 3:PEN 1                          [1137]
430 LOCATE 1,20:PRINT CHR$(150)STRING$(38, [2673]
154)CHR$(156)
```

Listing Apfelmann

```
440 LOCATE 1,25:PRINT CHR$(147)STRING$(38, [3878]
154)CHR$(153);
450 FOR a=21 TO 24:LOCATE 1,a:PRINT CHR$(1 [6288]
49);:LOCATE 40,a:PRINT CHR$(149);:NEXT
460 GOSUB 2390:menue=0                     [1169]
470 zeile$=" Esc: \  \  Modus: #  Monitor: [9812]
 \   \   Xmin:##.#########  Xmax:##.######
###  Ymin:##.#########  Ymax:##.#########
 Tiefe:####":pfeil$=CHR$(255)
480 GOSUB 2250:CLS                         [1255]
490 'Abfrage Menuepunkt                    [1321]
500 FOR a=0 TO 3:LOCATE 13,4+a*2:PRINT men [3706]
ue$(a,0);:NEXT
510 LOCATE 13,4+menue*2:PRINT CHR$(24)menu [5600]
e$(menue,0)CHR$(24);:altmenue=menue
520 a$=INKEY$:j=JOY(0)                     [546]
530 IF a$=""AND j=0 THEN 510               [1242]
540 IF a$=CHR$(240)OR(j AND 1)=1 THEN menu [2701]
e=(menue-1)AND 3
550 IF a$=CHR$(241)OR(j AND 2)=2 THEN menu [2260]
e=(menue+1)AND 3
560 IF a$=CHR$(224)OR(j AND 16)=16 THEN CL [2882]
S:GOTO 590
570 LOCATE 13,4+altmenue*2:PRINT menue$(al [2792]
tmenue,0);:GOTO 510
580 'Abfrage Untermenuepunkt               [1420]
590 GOSUB 2390:FOR a=1 TO 6                [2114]
600 LOCATE 7,a*2:IF menue$(menue,a)=""THEN [4930]
 620 ELSE PRINT menue$(menue,a)
610 NEXT                                   [350]
620 submenue=1:maxsub=a-1:a$=""            [2415]
630 GOSUB 2390:LOCATE 7,submenue*2:PRINT C [7686]
HR$(24)menue$(menue,submenue)CHR$(24);:hel
psub=submenue
640 a$=INKEY$:j=JOY(0):IF a$=""AND j=0 THE [1247]
N 640
650 IF a$=CHR$(240)OR(j AND 1)=1 THEN IF s [4680]
ubmenue=1 THEN submenue=maxsub ELSE submen
ue=submenue-1
660 IF a$=CHR$(241)OR(j AND 2)=2 THEN IF s [5299]
ubmenue=maxsub THEN submenue=1 ELSE submen
ue=submenue+1
670 IF a$=CHR$(224)OR(j AND 16)=16 THEN 70 [2404]
0
680 LOCATE 7,helpsub*2:PRINT menue$(menue, [4500]
helpsub);:GOTO 630
690 'Unterprogrammaufruf                   [1167]
700 IF submenue=maxsub THEN 470            [1959]
710 ON menue+1 GOTO 740,1380,1700,1790     [2465]
720 'Unterprogramme                        [916]
730 '1. Menuepunkt: Bild                   [2226]
740 ON submenue GOTO 760,830,960,1000,1190 [1430]
750 'Bild laden                            [1572]
760 GOSUB 900                              [832]
770 OPENIN file$+".dat"                    [1686]
780 INPUT#9,xmin!,xmax!,ymin!,ymax!,tmax%, [8283]
dx,dy,cx,cy,modus,zeile%,spalte%,endebild,
xleft!,timax%,monitor$:INPUT#9,esc$
790 LOAD file$+".pic",&C000                [1219]
800 CLOSEIN                                [752]
810 bild=1:GOTO 940                        [485]
820 'Bild speichern                        [2211]
830 IF bild=0 THEN WINDOW SWAP 2:GOTO 940  [2310]
ELSE GOSUB 900
840 OPENOUT file$+".dat"                   [1303]
850 PRINT#9,xmin!;xmax!;ymin!;ymax!;tmax%; [6152]
dx;dy;cx;cy;modus;zeile%;spalte%;endebild,
xleft!,timax%,monitor$:PRINT#9,esc$
860 CLOSEOUT                               [902]
870 SAVE file$+".pic",b,&C000,&4000        [2276]
880 GOTO 940                               [312]
890 'Abfrage Filename fuer Save/Load Bild  [2143]
900 WINDOW SWAP 2:CLS:PRINT:INPUT" Bitte g [6260]
eben Sie den Filenamen ein:     ",file$
910 IF LEN(file$)>8 OR INSTR(file$,".")>0  [6419]
OR file$=""THEN WINDOW SWAP 2:GOSUB 2250:G
OTO 630
920 file$="!"+file$:RETURN                 [852]
930 'Window#2 wieder herstellen            [1832]
940 WINDOW SWAP 2:GOSUB 2250:GOTO 630      [2680]
```

Listing Apfelmann

```
950 'Bild loeschen                               [1746]
960 IF bild=0 THEN 630 ELSE GOSUB 2280          [626]
970 IF a$="J"THEN CALL &BC06,&C0:CLG 0:CAL      [3008]
L &BC06,&40:bild=0:endebild=1
980 GOTO 940                                    [312]
990 'Ausschnitt vergroessern                    [881]
1000 IF bild=0 THEN WINDOW SWAP 2:CLS:PRIN      [7619]
T:PRINT CHR$(7)" Ausschnitt vergroessern n
icht moeg-  lich !";:CALL &BB06:GOTO 940
1010 CALL &BC06,&C0:MODE modus:GOSUB 2310:      [3476]
PRINT CHR$(23)"1";
1020 x=320:y=200:xx=x:yy=y:PLOT-4,-4,15         [1649]
1030 klick=0                                    [884]
1040 MOVE x,y:xx=x:yy=y:TAG:PRINT pfeil$;:      [4531]
MOVE x,y
1050 IF klick=1 THEN PLOT savx,savy:DRAW x      [4158]
,savy:DRAW savx,y:DRAW savx,savy
1060 a$=INKEY$:j=JOY(0):IF a$=""AND j=0 TH      [1291]
EN 1060
1070 IF INKEY(0)>-1 OR(j AND 1)=1 THEN IF       [3280]
y<397 THEN y=y+2
1080 IF INKEY(2)>-1 OR(j AND 2)=2 THEN IF       [2984]
y>0 THEN y=y-2
1090 IF INKEY(8)>-1 OR(j AND 4)=4 THEN IF       [3854]
x>0 THEN x=x-FNschritt
1100 IF INKEY(1)>-1 OR(j AND 8)=8 THEN IF       [3139]
x<639 THEN x=x+FNschritt
1110 IF INKEY(9)>-1 OR(j AND 16)=16 THEN I      [3962]
F klick=1 THEN 1150 ELSE savx=x:savy=y:kli
ck=1
1120 IF a$=CHR$(13)AND klick=1 THEN korr=1      [2732]
:GOTO 1150
1130 IF klick=1 THEN PLOT savx,savy:DRAW x      [5241]
x,savy:DRAW xx,yy:DRAW savx,yy:DRAW savx,s
avy
1140 MOVE xx,yy:PRINT pfeil$;:GOTO 1040         [1374]
1150 IF x=savx OR y=savy THEN CALL &BC06,&      [5073]
40:MODE 1:INK 1,14:INK 2,20:INK 3,26:GOTO
630
1160 ymin!=ymin!+savy*dy/2:xmax!=xmin!+x*d      [10491]
x/FNschritt:xmin!=xmin!+savx*dx/FNschritt:
IF korr=1 THEN ymax!=FNymax:ymin!=FNymin:x
max!=FNxmax:xmin!=FNxmin ELSE ymax!=ymin!+
y*dy/2
1170 CLG:endebild=1:esc$="Ja" 'Berechneten      [4492]
 Ausschnitt gleich zeichnen
1180 'Apfelmaennchen zeichnen                   [1468]
1190 CALL &BC06,&C0:MODE modus:GOSUB 2310       [1833]
1200 IF esc$="Nein"THEN POKE &B939,&C9          [1699]
1210 maxfarb=ABS((modus-2)^2*4-1)               [1544]
1220 IF endebild=1 THEN dx=FNdx:dy=FNdy:cx      [9289]
=xmin!:cy=ymin!:bild=1:zeile%=0:spalte%=0:
endebild=0:xleft!=xmin!:timax%=tmax% ELSE
cx=cx-dx
1230 FOR y%=zeile% TO 399 STEP 2                [1510]
1240 FOR x%=spalte% TO 639 STEP FNschritt       [2529]
1250 tiefe%=0:xwert=0:ywert=0:xquad=0:yqua      [4016]
d=0
1260 WHILE(tiefe%<timax%)AND(xquad+yquad<6      [3388]
)
1270 ywert=2*xwert*ywert+cy                     [1543]
1280 xwert=xquad-yquad+cx                       [1939]
1290 xquad=xwert^2                              [1481]
1300 yquad=ywert^2                              [1792]
1310 tiefe%=tiefe%+1:WEND                       [1418]
1320 IF tiefe%=timax% THEN PLOT x%,y%,0:cx      [4107]
=cx+dx:GOTO 1340
1330 cx=cx+dx:farbe%=tiefe%MOD maxfarb+1:P      [2812]
LOT x%,y%,farbe%
1340 IF INKEY$<>""THEN 1360                     [934]
1350 NEXT:spalte%=0:cx=xleft!:cy=cy+dy:NEX      [5121]
T:x%=0:y%=0:endebild=1
1360 spalte%=x%:zeile%=y%:POKE &B939,&F3:C      [6357]
ALL &BC06,&40:MODE 1:INK 1,14:INK 2,20:INK
 3,26:GOTO 630
1370 '2. Menuepunkt: Drucker                    [927]
1380 ON submenue GOTO 1400,1540,1680            [859]
1390 'Anpassung anderer Drucker                 [2346]
1400 WINDOW SWAP 2:a$="17/216 Zoll Zeilena      [4111]
bstand":GOSUB 1480:abstand$=b$
1410 a$="1/216 Zoll Zeilenabstand":GOSUB 1      [4688]
480:abstand1$=b$
1420 a$="Grafik in dopp.Dichte(800 Pkt/Zei      [4955]
le)":GOSUB 1480:abstand2$=b$
1430 a$="1/8 Zoll Zeilenabstand":GOSUB 148      [4389]
0:abstand3$=b$
1440 a$="Seitenvorschub":GOSUB 1480:abstan      [2524]
d4$=b$
```

Listing Apfelmann

```
1450 a$="Wagenruecklauf und Zeilenvorschub      [5522]
":GOSUB 1480:abstand5$=b$
1460 a$="Druckerreset":GOSUB 1480:abstand6      [3423]
$=b$
1470 GOSUB 2560:GOTO 940                        [2220]
1480 CLS:PRINT" Bitte geben Sie die Anzahl      [10164]
 der Steu-  erzeichen der Escape-Sequenz":
PRINT" "CHR$(24)a$CHR$(24)" ein: ";:INPUT"
",anzahl
1490 IF anzahl=0 THEN 940 ELSE b$=""            [1244]
1500 FOR x=1 TO anzahl:PRINT" Wert "x". Ze      [3860]
ichen: ";:INPUT"",wert
1510 b$=b$+CHR$(wert):NEXT                      [474]
1520 RETURN                                     [555]
1530 'Hardcopy                                  [320]
1540 |BUSY,@busy%:IF (bild AND busy%)=0 TH      [4452]
EN 630 'Test ob Drucker bereit
1550 IF endebild=0 THEN WINDOW SWAP 2:CLS:      [10861]
PRINT:PRINT" Moechten Sie das unfertige Bi
ld     schon ausdrucken ? (J/N) ";:GOSUB
 2290:IF a$<>"J"THEN WINDOW SWAP 2:GOSUB 2
250:GOTO 630
1560 CALL &BC06,&C0:MODE modus:GOSUB 2310       [1833]
1570 adr=&8000:length=0 'Beginn Speicher d      [5922]
er Druckervariablen; Ablegen Adressen der
Druckervariablen im Speicher fuer Hardcopy
1580 POKE &A269,0:POKE &A26A,&80 '17/216 Z      [3869]
oll Zeilenabstand
1590 GOSUB 2530:POKE &A26B,adr AND 255:POK      [6280]
E &A26C,INT(adr/256)+(256 AND INT(adr/256)
<0) '1/216 Zoll Zeilenabstand
1600 GOSUB 2530:POKE &A26D,adr AND 255:POK      [10122]
E &A26E,INT(adr/256)+(256 AND INT(adr/256)
<0) 'Grafik in doppelter Dichte (960 Punkt
e/Zeile) mit 800 Punkte/Zeile
1610 GOSUB 2530:POKE &A26F,adr AND 255:POK      [7168]
E &A270,INT(adr/256)+(256 AND INT(adr/256)
<0) '1/8 Zoll Zeilenabstand
1620 GOSUB 2530:POKE &A271,adr AND 255:POK      [4514]
E &A272,INT(adr/256)+(256 AND INT(adr/256)
<0) 'Wagenruecklauf und Zeilenvorschub
1630 GOSUB 2530:POKE &A273,adr AND 255:POK      [5659]
E &A274,INT(adr/256)+(256 AND INT(adr/256)
<0) 'Druckerreset
1640 |HARDCOPY:GOSUB 2560                       [2023]
1650 PRINT#8,"Xmin:  ";xmin!:PRINT#8,"Xmax      [9833]
:   ";xmax!:PRINT#8,"Ymin:   ";ymin!:PRINT#8
,"Ymax:   ";ymax!:PRINT#8,"Tiefe:  ";tmax%;a
bstand4$abstand6$; 'Ausdruck von Xmin,Xmax
,Ymin,Ymax,Tiefe, Seitenvorschub und Druck
erreset
1660 CALL &BC06,&40:MODE 1:INK 1,14:INK 2,      [3565]
20:INK 3,26:GOTO 630
1670 'Druckerparameter speichern                [3046]
1680 SAVE"!apfelman.dat",b,&8000,length:GO      [3493]
TO 630
1690 '3. Menuepunkt: Programm                   [1073]
1700 ON submenue GOTO 1720,1760                 [744]
1710 'Ende                                      [715]
1720 GOSUB 2280                                 [851]
1730 IF a$="J"THEN MODE modus:CALL &BC06,&      [3031]
C0:GOSUB 2310:END
1740 GOTO 940                                   [312]
1750 'Neustart                                  [759]
1760 GOSUB 2280                                 [851]
1770 IF a$="J"THEN a=PEEK(&BDEC)+256*PEEK(      [10769]
&BDED)-23:POKE &BDEC,a AND 255:POKE &BDED,
INT(a/256):MEMORY &8FFF:CLOSEOUT:CLOSEIN:M
EMORY &A67B:RUN ELSE 940
1780 '4. Menuepunkt: Sonstiges                  [1610]
1790 ON submenue GOTO 1810,1840,1870,2000,      [1037]
2220
1800 'Umschalten Farb-/Gruenmonitor             [1778]
1810 IF monitor$="Gruen"THEN monitor$="Far      [3424]
be"ELSE monitor$="Gruen"
1820 GOSUB 2250:GOTO 630                        [1289]
1830 'Esc ein/aus                               [825]
1840 IF esc$="Ja"THEN esc$="Nein"ELSE esc$      [2009]
="Ja"
1850 GOSUB 2250:GOTO 630                        [1289]
1860 'Darstellungsmodus                         [906]
1870 CLS:FOR a=0 TO 2:LOCATE 4,4+a*2:PRINT      [4276]
 modus$(a):NEXT:newmodus=0
1880 LOCATE 4,4+newmodus*2:PRINT CHR$(24)m      [5263]
odus$(newmodus)CHR$(24);:helpmodus=newmodu
s
1890 a$=INKEY$:j=JOY(0):IF a$=""AND j=0 TH      [1428]
EN 1890
```

Listing Apfelmann

Auf die Plätze... fertig ... los

Mitmachen und Geld gewinnen bei unserem Wettbewerb

Sie halten gerade das neue CPC-Sonderheft in den Händen. Das heißt, Sie besitzen einen CPC und wollen Ihren Computer mit Software füttern. Und genau mit diesem Heft können Sie dem ewig hungrigen CPC eine Menge Futter zuführen.

Eine Spezialität in diesem Heft sind unsere Mini-Programme, die besonders klein und unglaublich pfiffig sind.

Wenn Sie wissen wollen, was die findigen Leser und Leserinnen programmiert haben, dann schauen Sie doch gleich in die Rubrik "Tips und Tricks". Dort finden Sie eine Menge von Programmen, die Ihnen bei der Bedienung und dem Programmieren des CPC weiterhelfen, und einen Einstieg in die Computersprache BASIC.

Die Programme erstrecken sich von der einfachen Menüsteuerung bis hin zum komplizierten Apfelmännchen.

Doch auch der Spielefreund bekommt was geboten, vom witzigen Galgenspiel bis hin zu einem komplexen Labyrinth-Spiel.

Wir finden, diese Leistung muß besonders honoriert werden. Deshalb machen wir Ihnen ein Angebot: Senden auch Sie uns ein Programm ein, das die unten angegebenen Bedingungen erfüllt. Die Belohnung für eine Veröffentlichung beträgt einhundert Deutsche Mark.

Bedingungen:

Auch Sie können einhundert Deutsche Mark gewinnen, wenn folgende Punkte beachtet werden:

1. Senden Sie uns das Programm (maximal zwei!) auf einer Kassette oder Diskette zu – wird bei Nichtgefallen zurückgeschickt.
2. Speichern Sie Ihre Bedienungsanleitung für das Programm im ASCII-Format ab. Entweder mit einer Textverarbeitung, zum Beispiel ConText, oder, wenn keine Textverarbeitung vorliegt, in einem BASIC-Programm mit REM-Zeilen. Hierbei ist es wichtig, daß das Programm mit ",A" abgespeichert wurde.
Beispiel: SAVE"Prg-Text.TXT",A
Wenn die Möglichkeit besteht, sollte der Text auch ausgedruckt beiliegen.
3. Der Rechnertyp (464/664/6128) sollte mit angegeben sein.
4. Wohin schicke ich die Programme ein? Ganz einfach! Schreiben Sie an den:

DMV-Verlag
Stichwort: 100,– DM
Redaktion PC Amstrad
Postfach 250
3440 Eschwege

Die Programme werden bei Gefallen in der monatlich erscheinenden PC Amstrad International veröffentlicht.
Viel Glück wünscht Ihre Redaktion PC Amstrad.

```
1900 IF a$=CHR$(240)OR(j AND 1)=1 THEN IF  [4670]
newmodus=0 THEN newmodus=2 ELSE newmodus=n
ewmodus-1
1910 IF a$=CHR$(241)OR(j AND 2)=2 THEN IF  [4930]
newmodus=2 THEN newmodus=0 ELSE newmodus=n
ewmodus+1
1920 IF a$=CHR$(224)OR(j AND 16)=16 THEN 1 [2461]
940
1930 LOCATE 4,4+helpmodus*2:PRINT modus$(h [2876]
elpmodus);:GOTO 1880
1940 IF modus<>newmodus THEN 1950 ELSE mod [4903]
us=newmodus:CLS:GOSUB 2250:GOTO 590
1950 IF modus=2 THEN IF newmodus=1 THEN dx [3769]
=dx/2 ELSE dx=dx/4
1960 IF modus=1 THEN IF newmodus=0 THEN dx [2513]
=dx*2 ELSE dx=dx/2
1970 IF modus=0 THEN IF newmodus=1 THEN dx [2641]
=dx*2 ELSE dx=dx*4
1980 modus=newmodus:CLS:GOSUB 2250:GOTO 59 [2864]
0
1990 'Werte veraendern                     [1165]
2000 CLS:FOR a=0 TO 4:LOCATE 4,2+a*2:PRINT [3612]
 wert$(a);:NEXT
2010 wert=0                                [533]
2020 LOCATE 4,2+wert*2:PRINT CHR$(24)wert$ [2353]
(wert)CHR$(24);:altwert=wert
2030 a$=INKEY$:j=JOY(0):IF a$=""AND j=0 TH [1080]
EN 2030
2040 IF a$=CHR$(240)OR(j AND 1)=1 THEN IF  [2662]
wert=0 THEN wert=4 ELSE wert=wert-1
2050 IF a$=CHR$(241)OR(j AND 2)=2 THEN IF  [3813]
wert=4 THEN wert=0 ELSE wert=wert+1
2060 IF a$=CHR$(224)OR(j AND 16)=16 THEN G [3051]
OSUB 2090
2070 IF a$=CHR$(13)THEN 2160               [692]
2080 LOCATE 4,2+altwert*2:PRINT wert$(altw [3120]
ert);:GOTO 2020
2090 LOCATE 10,2+wert*2:INPUT":  ",wert2   [3134]
2100 IF wert=0 THEN xmin!=wert2            [1790]
2110 IF wert=1 THEN xmax!=wert2            [2151]
2120 IF wert=2 THEN ymin!=wert2            [2305]
2130 IF wert=3 THEN ymax!=wert2            [1301]
2140 IF wert=4 THEN tmax%=wert2            [1189]
2150 GOSUB 2250:LOCATE 10,2+wert*2:PRINT S [3042]
PC(20);:GOTO 2020
2160 WINDOW SWAP 2:CLS:PRINT:PRINT" Korrek [7611]
tur der eingegebenen Werte ?   (J/N) "CHR
$(143)CHR$(8);
2170 GOSUB 2290:PRINT CHR$(8);:IF a$<>"J"A [5651]
ND a$<>"N"THEN PRINT" "CHR$(7)CHR$(8);:GOT
O 2170
2180 IF a$="N"THEN 2200                    [973]
2190 ymax!=FNymax:ymin!=FNymin:xmax!=FNxma [3480]
x
2200 WINDOW SWAP 2:GOSUB 2250:CLS:GOTO 590 [4065]
2210 'Bild anzeigen                        [851]
2220 CALL &BC06,&C0:MODE modus:GOSUB 2310  [1833]
2230 CALL &BB03:CALL &BB06:CALL &BC06,&40: [4862]
MODE 1:INK 1,14:INK 2,20:INK 3,26:GOTO 630
2240 'Werteanzeige in Window#2             [2589]
2250 CLS#2:PRINT#2,USING zeile$;esc$;modus [6231]
;monitor$;xmin!;xmax!;ymin!;ymax!;tmax%;
2260 RETURN                                [555]
2270 'Sicherheitsabfrage                   [1051]
2280 WINDOW SWAP 2:CLS:PRINT:PRINT" Sind S [4652]
ie sicher ? (J/N) "CHR$(143)CHR$(8);
2290 a$="":WHILE a$="":a$=UPPER$(INKEY$):W [3096]
END:PRINT a$;:RETURN
2300 'Farben setzen                        [1432]
2310 IF modus=2 THEN INK 1,26:RETURN ELSE  [5060]
IF modus=1 THEN INK 1,14:INK 2,20:INK 3,26
:RETURN
2320 IF monitor$="Gruen"THEN RESTORE 2340  [3487]
ELSE RESTORE 2350
2330 FOR a=1 TO 15:READ b:INK a,b:NEXT:RET [1974]
URN
2340 DATA 2,4,8,10,12,14,16,18,20,21,22,23 [1536]
,24,25,26
2350 DATA 6,15,7,2,5,11,14,20,23,22,21,19, [2186]
24,25,26
2360 'ESC gedrueckt                        [487]
2370 RETURN                                [555]
2380 'Windows definieren                   [933]
2390 WINDOW#1,1,40,1,4:PAPER#1,3:PEN#1,1   [3106]
2400 WINDOW#2,2,39,21,24:PAPER 0:PEN 2:PAP [3394]
ER#2,0:PEN#2,1
2410 WINDOW 2,39,6,18:PAPER 0:PEN 3        [1264]
2420 RETURN                                [555]
2430 'Einlesen Druckervariablen            [2505]
```

Listing Apfelmann

```
2440 adr=&8000:length=0                        [783]
2450 GOSUB 2530:abstand$=b$                    [2054]
2460 GOSUB 2530:abstand1$=b$                   [1980]
2470 GOSUB 2530:abstand2$=b$                   [1964]
2480 GOSUB 2530:abstand3$=b$                   [2396]
2490 GOSUB 2530:abstand5$=b$                   [2044]
2500 GOSUB 2530:abstand6$=b$                   [2284]
2510 GOSUB 2530:abstand4$=b$                   [2380]
2520 RETURN                                    [555]
2530 b$=""                                     [53]
2540 x=PEEK(adr):adr=adr+1:length=length+1    [7007]
:IF x<>255 THEN b$=b$+CHR$(x):GOTO 2540 EL
SE RETURN
2550 'Ablegen Druckervariablen                 [2404]
2560 adr=&8000:length=0                        [783]
2570 b$=abstand$:GOSUB 2650                    [2220]
2580 b$=abstand1$:GOSUB 2650                   [1433]
2590 b$=abstand2$:GOSUB 2650                   [1608]
2600 b$=abstand3$:GOSUB 2650                   [1426]
2610 b$=abstand5$:GOSUB 2650                   [1421]
2620 b$=abstand6$:GOSUB 2650                   [1483]
2630 b$=abstand4$:GOSUB 2650                   [1779]
2640 RETURN                                    [555]
2650 FOR x=1 TO LEN(b$):POKE adr,ASC(MID$(    [7600]
b$,x,1)):adr=adr+1:length=length+1:NEXT:PO
KE adr,255:adr=adr+1:length=length+1:RETUR
N
```

```
10 '************************************ [1668]
*
20 '* APFELMAN.BIN - DATA-Lader von 'CPC'  [1729]
*
30 '************************************ [1668]
*
40 '                                       [117]
50 DATA A000,01,09,A0,21,1E,A0,C3,D1,BC,11 [3355]
,A0,C3,22,A0,C3,31,B740
60 DATA A010,A0,42,55,53,D9,48,41,52,44,43 [3362]
,4F,50,D9,00,00,00,10A0
70 DATA A020,00,00,FE,01,C0,CD,2E,BD,38,04 [2378]
,3E,01,12,C9,AF,12,1D32
80 DATA A030,C9,B7,C0,AF,32,67,A2,CD,11,BC [4172]
,32,68,A2,2A,73,A2,17E6
90 DATA A040,CD,90,A1,21,01,00,22,63,A2,2B [2435]
,11,80,02,EB,B7,ED,CC8B
100 DATA A050,52,38,37,28,35,2A,63,A2,2B,2 [3716]
2,63,A2,2A,6B,A2,CD,1E17
110 DATA A060,90,A1,CD,90,A0,2A,69,A2,CD,9 [2768]
0,A1,CD,90,A0,CD,09,6596
120 DATA A070,BB,38,17,2A,63,A2,23,22,63,A [3210]
2,3A,67,A2,2F,32,67,29CD
130 DATA A080,A2,2A,63,A2,11,06,00,19,18,B [2784]
C,2A,6F,A2,C3,90,A1,6E70
140 DATA A090,2A,6D,A2,CD,90,A1,21,01,00,2 [2443]
2,65,A2,2B,11,90,01,992D
150 DATA A0A0,EB,B7,ED,52,38,21,28,1F,AF,3 [2924]
2,5D,A2,21,5E,A2,06,90F3
160 DATA A0B0,04,77,23,10,FC,CD,CD,A0,21,5 [3188]
E,A2,CD,90,A1,2A,65,57BA
170 DATA A0C0,A2,11,02,00,19,18,D2,2A,71,A [2465]
2,C3,90,A1,F3,3A,68,21EC
180 DATA A0D0,A2,B7,28,58,FE,01,28,28,06,0 [3135]
6,ED,5B,63,A2,2A,65,D74A
190 DATA A0E0,A2,C5,D5,E5,CD,F0,BB,21,5D,A [2830]
2,46,CB,00,B0,77,E1,4FCF
200 DATA A0F0,D1,13,C1,10,EC,21,5E,A2,06,0 [2963]
4,77,23,10,FC,FB,C9,96ED
210 DATA A100,ED,5B,63,A2,2A,65,A2,D5,E5,C [2689]
D,B0,A1,11,5E,A2,01,0802
220 DATA A110,04,00,ED,B0,E1,D1,13,13,D5,E [2721]
5,CD,B0,A1,CD,A1,A1,9FAF
230 DATA A120,E1,D1,13,13,CD,B0,A1,CD,A1,A [2039]
1,FB,C9,ED,5B,63,A2,DA12
240 DATA A130,2A,65,A2,3A,67,A2,B7,20,2C,D [3825]
5,E5,CD,C8,A1,11,5E,6993
250 DATA A140,A2,01,04,00,ED,B0,E1,D1,13,1 [2401]
3,13,13,CD,C8,A1,11,715E
260 DATA A150,5E,A2,06,04,1A,07,07,07,07,B [3440]
6,0F,0F,E6,3F,12,13,EDEC
270 DATA A160,23,10,F1,FB,C9,D5,E5,CD,C8,A [3238]
1,11,5E,A2,06,04,7E,45FE
280 DATA A170,07,07,07,07,E6,3F,12,13,23,1 [3482]
0,F4,E1,D1,13,13,CD,283A
290 DATA A180,C8,A1,11,5E,A2,06,04,1A,B6,1 [3244]
2,13,23,10,F9,FB,C9,C7C5
```

Listing Apfelmann

```
300 DATA A190,7E,FE,FF,C8,CD,9B,A1,23,18,F [4646]
6,2B,CD,2B,BD,30,FB,BE11
310 DATA A1A0,C9,11,5E,A2,06,04,1A,07,07,B [3270]
6,12,13,23,10,F7,C9,1E6E
320 DATA A1B0,CD,F0,BB,B7,21,0D,A2,C8,21,1 [3581]
1,A2,FE,01,C8,21,15,2C59
330 DATA A1C0,A2,FE,02,C8,21,19,A2,C9,CD,F [2371]
0,BB,21,1D,A2,11,04,ECD4
340 DATA A1D0,00,B7,C8,19,FE,01,C8,19,FE,0 [4884]
2,C8,19,FE,03,C8,19,4C56
350 DATA A1E0,FE,04,C8,19,FE,05,C8,19,FE,0 [3851]
6,C8,19,FE,07,C8,19,9D15
360 DATA A1F0,FE,08,C8,19,FE,09,C8,19,FE,0 [3182]
A,C8,19,FE,0B,C8,19,AE26
370 DATA A200,FE,0C,C8,19,FE,0D,C8,19,FE,0 [2442]
E,C8,19,C9,00,00,00,BF3A
380 DATA A210,00,01,01,02,02,01,02,01,02,0 [2215]
3,03,03,03,00,00,00,5088
390 DATA A220,00,00,04,00,00,08,00,02,00,0 [2882]
8,04,02,01,0C,02,01,A09D
400 DATA A230,01,0A,02,0A,02,0A,0A,0A,0A,0 [3463]
3,09,0C,06,03,03,0C,C214
410 DATA A240,0C,0F,00,0F,00,0A,05,0A,05,0 [2740]
5,05,05,0F,0F,09,09,1B6A
420 DATA A250,0F,0F,0A,0F,0A,0F,0B,0B,0F,0 [2407]
F,0F,0F,0F,00,00,00,842D
430 DATA A260,00,00,FF,00,00,91,01,00,01,0 [2149]
0,80,04,80,08,80,0D,B3F0
440 DATA A270,80,12,80,15,80,00,00,00,00,0 [2072]
0,00,00,00,00,00,00,D051
450 DATA *ENDE*                            [502]
460 adr=&A000:zeile=50:MEMORY adr-1        [1876]
470 READ d$:IF d$="*ENDE*"THEN 580         [1076]
480 pr=0                                   [117]
490 FOR i=1 TO 16                          [317]
500 READ a$:a=VAL("&"+a$)                  [883]
510 POKE adr,a:adr=adr+1                   [717]
520 pr=pr*2:IF pr>65535 THEN pr=pr-65535   [2485]
530 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+655 [1890]
36
540 NEXT                                   [350]
550 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THE [2610]
N pr2=pr2+65536
560 IF pr<>pr2 THEN PRINT"Pruefsummenfehle [3615]
r in Zeile";zeile:STOP
570 zeile=zeile+10:GOTO 470                [2018]
580 SAVE"!apfelman.bin",b,&A000,&277       [2182]
590 PRINT d$:END                           [346]
b$,x,1)):adr=adr+1:length=length+1:NEXT:PO
KE adr,255:adr=adr+1:length=length+1:RETUR
N
```

```
10 abstand$=CHR$(27)+"3"+CHR$(17)                  [3640]
 '17/216 Zoll Zeilenabstand
20 abstand1$=CHR$(27)+"3"+CHR$(1)                  [2728]
 '1/216 Zoll Zeilenabstand
30 abstand2$=CHR$(27)+"L"+CHR$(32)+CHR$(3)         [6555]
 'Grafik in doppelter Dichte (960 Punkte/Z
eile) mit 800 Punkten/Zeile
40 abstand3$=CHR$(27)+"0"                          [2406]
 '1/8 Zoll Zeilenabstand
50 abstand4$=CHR$(12)                              [2311]
 'Seitenvorschub
60 abstand5$=CHR$(13)+CHR$(10)                     [4025]
 'Wagenrueecklauf und Zeilenvorschub
70 abstand6$=CHR$(27)+"@"                          [2118]
 'Druckerreset
80 adr=&8000:length=0                              [783]
90 b$=abstand$:GOSUB 210                           [2061]
100 b$=abstand1$:GOSUB 210                         [1431]
110 b$=abstand2$:GOSUB 210                         [1615]
120 b$=abstand3$:GOSUB 210                         [1304]
130 b$=abstand5$:GOSUB 210                         [1468]
140 b$=abstand6$:GOSUB 210                         [1564]
150 b$=abstand4$:GOSUB 210                         [1736]
160 PRINT"Datentraeger bereitmachen und Ta         [5005]
ste druecken: ";
170 CALL &BB81:CALL &BB03:CALL &BB06:CALL          [1956]
&BB84
180 SAVE"!apfelman.dat",b,&8000,length             [2878]
190 END                                            [110]
200 'Ablegen der Druckervariablen im Speic         [4366]
her
210 FOR x=1 TO LEN(b$):POKE adr,ASC(MID$(b         [6583]
$,x,1)):adr=adr+1:length=length+1:NEXT:POK
E adr,255:length=length+1:adr=adr+1:RETURN
```

Listing Apfelmann

Bestellen mit Freu(n)den

Sammelbestellungen einmal ganz anders!

Das Programm "Sammelbestellung" ermöglicht dem Sammelbesteller die Bearbeitung von Sammelbestellungen und Zurücklieferungen, sowie die Kontoführungen. Sammelbestellungen werden im allgemeinen unter Verwandten, Freunden und Bekannten durchgeführt. Daher kann eine speicherplatzfressende Adressenverwaltung entfallen.

Wenn Bestellungen bei mehreren Versandhäusern getätigt werden, sollten die Namen der Versandhäuser im Programm geändert werden. Das Programm wird dann unter dem Namen des Versandhauses auf Diskette abgespeichert. Durch Änderung des Anfangsbuchstaben der einzelnen Dateien ('Q'sammel.dat, 'Q'a.dat für "QUELLE" bzw. "O"sammel.dat, Oa.dat für "OTTO") ist eine Sicherung der Dateien mehrerer Versandhäuser auf einer Diskette möglich. Kassettenbearbeitung ist nur unter großem Aufwand durchzuführen. Wenn erforderlich, sollte das Programm in den dafür nötigen Zeilen geändert werden. Die Sammel- und Mitbestellerdateien müssen aber getrennt auf verschiedenen Kassetten gesichert werden. Alle erforderlichen Daten werden auf dem Monitor angezeigt, können aber mit einem Drucker, DMP 2000 zum Beispiel, ausgedruckt werden.

Wie wird's nun gemacht?

1. Abfrage des Tagesdatums
Nach dem Starten des Programmes wird nach dem Datum des jeweiligen Tages gefragt. Beispiel = 1.1.1989.

2. Sammelbestellerdaten laden
Bei Neustart des Programmes werden die Daten des Sammelbestellers geladen. Nach Rücksprüngen an den Anhang des Programmes kann ein nochmaliges Laden entfallen. Ist die Sammelbestellerdatei geladen, werden die Daten angezeigt.

3. Liste der Mitbesteller
Die Liste der Mitbesteller kann jetzt angezeigt werden. Wird bei diesem Programmpunkt die Frage verneint, springt das Programm zur Anzeige des Disketteninhaltes. So kann am Anfang der Bearbeitung der Disketteninhalt eingesehen werden. Bei der Anzeige der Mitbesteller wird nach dem zu bearbeitenden gefragt. Es brauchen nur die beiden Buchstaben (Strings), die zwischen den Trennstrichen stehen, eingegeben zu werden. Dieser String dient dann als Name der abzuspeichernden Datei.

4. Mitbestellerdaten laden
Bei der Erstbestellung eines Mitbestellers muß dieser Programmpunkt übersprungen werden. Anderenfalls werden die Daten geladen und angezeigt.

5. Bestellungen eingeben
Neue Bestellungen werden in diesem Programmpunkt eingegeben. Der Computer überprüft vor einer neuen Eingabe, ob aus vorherigen Bestellungen noch Außenstände vorhanden sind. Ist dieses der Fall, gibt er weitere Anweisungen. Bei einer Neueingabe wird zuerst nach dem Vor- und Nachnamen des Mitbestellers gefragt. Anschließend können die erforderlichen Bestelldaten eingegeben werden. Bei Ratenzahlungen errechnet der Computer die sich ergebenden Ratenbeträge. Anfallende Umsatzbeteiligungen werden dem Konto des Sammelbestellers gutgeschrieben.

6. Retouren eingeben
Etwaige Zurücklieferungen werden hier eingegeben und mit der eingegebenen Bestellung verrechnet.

7. Gutschrift von Retourzinsen
Bei Zurücklieferungen in Verbindung mit Ratenzahlungen werden vom Versandhaus Retourzinsen gutgeschrieben. Diese werden nach der Eingabe auf den Konten verbucht.

8. Einzelbeträge einzahlen
Es müssen hier alle bezahlten Beträge, außer Ratenbeträge, eingegeben werden.

9. Ratenbeträge einzahlen
Bei Einzahlung eines Ratenbetrages wird die Summe den Konten gutgeschrieben, und die Anzahl der bezahlten Raten um 1 erhöht. Werden überhöhte Ratenzahlungen geleistet, wird bei vorzeitiger Tilgung der Ratensumme auch die Anzahl der Raten angeglichen.

10. Mitbestellerdaten anzeigen
Nach Eingabe aller Daten werden diese angezeigt und können überprüft werden.

11. Sammelbestellerdaten anzeigen.
Siehe unter Nummer 10.

12. Daten abspeichern
Alle wichtigen Daten können abgespeichert werden. Auf dem Datenträger werden zuerst die Mitbestellerdaten, danach automatisch die Sammelbestellerdaten gesichert. Sicherheitskopien beider Daten können ebenfalls erstellt werden.

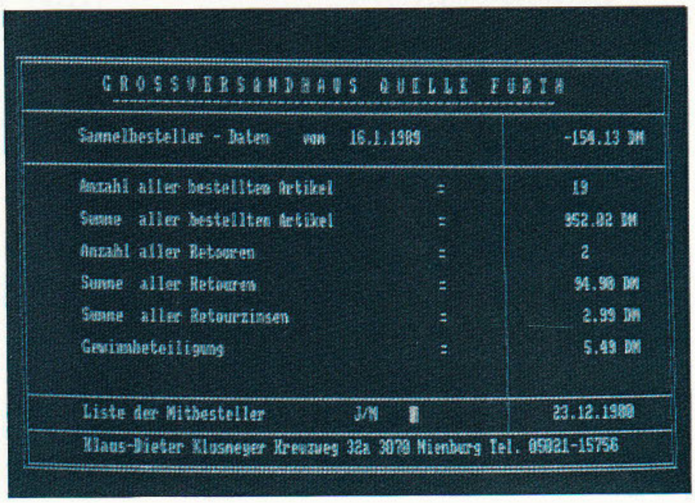
13. Daten ausdrucken
Der Sammelbesteller kann alle Daten, die Sammelbestellerdaten und die Mitbestellerdaten, ausdrucken lassen.

14. Disketteninhalt anzeigen
Nach dem Löschen des Bildschirminhalts wird der Inhalt der Diskette angezeigt. Durch Drücken einer Taste wird an den Anfang des Programmes zurückgekehrt, aber alle Daten bleiben erhalten.

15. Programm beenden
Nach einer Sicherheitsüberprüfung (alle Daten gesichert?) kann das Programm beendet werden.

16. Sammelbestellerdatei erstellen
Bei Erstbeginn des Programmes muß die Sammelbestellerdatei erstellt werden. Dazu muß in Zeile 570 das REM-Zeichen entfernt und das Programm gestartet werden. Nach den Anweisungen diese Programmteiles wird die Datei auf dem Datenträger erstellt.

(Klaus-Dieter Klusmeyer/cd)



Die Verwaltung von Sammelbestellungen ist jetzt keine Qual mehr

Variablenliste

datum$: Datum des Tages der Bearbeitung
datuml$: Datum der letzten Bearbeitung
allkonto: Kontostand des Sammelbestellers
allstueck: Anzahl aller bestellten Artikel
allpreis: Summe aller bestellten Artikel
allretur: Anzahl aller Retouren
allretursumme: Summe aller Retouren
allreturzins: Summe der Retourzinsen
allgewinn: Summe der Umsatzbeteiligung
mitbest$: String der Mitbesteller
name1$: Vorname der Mitbesteller
name2$: Nachname der Mitbesteller
stueck: Anzahl der bestellten Artikel eines Mitbestellers
gesstueck: Anzahl aller bestellten Artikel eines Mitbestellers
retur: Anzahl der Retouren eines Mitbestellers
gesretur: Anzahl aller Retouren eines Mitbestellers
retursumme: Summe der Retouren eines Mitbestellers
gesretursumme: Summe aller Retouren eines Mitbestellers
preis: Summe der bestellten Artikel eines Mitbestellers
gespreis: Summe aller bestellten Artikel eines Mitbestellers
raten: Anzahl der Raten
bezraten: Anzahl der bezahlten Raten
ratzins: Summe der Ratenzinsen
monat: Monat der ersten Ratenzahlung
retgewinn: Gewinnsumme bei Retouren
behpreis: Summe der behaltenen Artikel
returzins: Gutschrift von Retourzinsen
betrag: Bezahlter Betrag eines Mitbestellers
ratenbetrag: Bezahlter Ratenbetrag eines Mitbestellers
versspes: Versandspesen je Bestellung
versspesen: Versandspesen aller Bestellungen
pruefung,jump, sprung: Sprungvariablen
j$: Ja/Nein Abfrage

für 464-664-6128

```
10 ' ************************************** [2556]
**************************************
20 ' *                                      [175]
                                     *
30 ' *              S A M M E L B E S T E [997]
 L L U N G E N                       *
40 ' *                                      [175]
                                     *
50 ' *       Ein Programm zur Bearbeitun [3070]
g von Sammelbestellungen             *
60 ' *                                      [175]
                                     *
70 ' *   Klaus-Dieter Klusmeyer             [1546]
                                     *
80 ' *                                      [175]
                                     *
90 ' ************************************** [2556]
**************************************
100 '                                       [117]
110 ' deutscher zeichensatz                 [1352]
120 '                                       [117]
130 SYMBOL AFTER 91                         [1213]
140 SYMBOL 91,&66,&18,&3C,&66,&7E,&66,&66, [2047]
&0  'AE
150 SYMBOL 92,&66,&3C,&66,&66,&66,&66,&3C, [2386]
&0  'OE
160 SYMBOL 93,&66,&44,&66,&66,&66,&66,&3C, [2324]
&0  'UE
170 SYMBOL 123,&CC,0,&78,&C,&7C,&CC,&76,&0 [2508]
 'ae
180 SYMBOL 124,&66,0,&3C,&66,&66,&66,&3C,& [2847]
0  'oe
190 SYMBOL 125,&66,0,&66,&66,&66,&66,&3E,& [1864]
0  'ue
200 SYMBOL 126,&3C,&66,&66,&6C,&66,&66,&6C [2854]
,&60       'scharfes s
210 KEY DEF 17,1,125,93                     [746]
220 KEY DEF 19,1,123,91                     [1140]
230 KEY DEF 26,1,124,92,64                  [765]
240 KEY DEF 24,1,126,94,0                   [1480]
250 '                                       [117]
260 ' bildschirmaufbau                      [1166]
270 '                                       [117]
280 sprung=0:jump=0:raten=0:bezraten=0      [2860]

290 CLS : MODE 2                            [460]
300 PLOT 1,1:DRAW 1,399: DRAW 639,399:DRAW [2779]
 639,1:DRAW 1,1
310 PLOT 5,5:DRAW 5,394: DRAW 635,394:DRAW [2430]
 635,5:DRAW 5,5
320 PLOT  6,350:DRAW 635,350                [1350]
330 PLOT 6,36:DRAW 635,36                   [757]
340 PLOT 6,70:DRAW 635,70                   [1069]
350 PLOT 475,36:DRAW 475,350                [1618]
360 PLOT  6,300:DRAW 635,300                [1267]
370 WINDOW #0,2,79,2,3                      [826]
380 WINDOW #1,2,59,5,6                      [1330]
390 WINDOW #2,61,79,5,6                     [1128]
400 WINDOW #3,2,59,8,20                     [1031]
410 WINDOW #4,61,79,8,20                    [1291]
420 WINDOW #5,2,59,22,22                    [1445]
430 WINDOW #6,61,79,22,22                   [1157]
440 WINDOW #7,2,79,24,24                    [1102]
```

Listing SAMMEL

```
450 LOCATE #0,10,1:PRINT #0,"G R O S S V E [4463]
 R S A N D H A U S   Q U E L L E   F ]R T
H"
460 LOCATE #0,11,2:PRINT #0,"------------- [4039]
----------------------------------------
"
470 CLS#7:LOCATE #7,7,1:PRINT #7,"Klaus-Di [3408]
eter Klusmeyer "
480 CLS#3                                  [371]
490 LOCATE #3,11,4:PRINT #3,"============ [3297]
=========================="
500 LOCATE #3,11,5:PRINT #3,"=             [2507]
                                    ="
510 LOCATE #3,11,6:PRINT #3,"=    S A M M E [3422]
 L B E S T E L L U N G              ="
520 LOCATE #3,11,7:PRINT #3,"=             [2344]
                                    ="
530 LOCATE #3,11,8:PRINT #3,"============= [3375]
=========================="
540 '                                      [117]
550 ' laufwerk b ist das dateilaufwerk     [2012]
560 '                                      [117]
570 '|B                                    [219]
580 '                                      [117]
590 ' sammelbestellerdatei erstellen       [2028]
600 '                                      [117]
610 'FOR i=1 TO 2000:NEXT i:GOSUB 4270:CLS [2129]
#3
620 '                                      [117]
630 CLS#1::CLS#2:CLS#4:CLS#5:CLS#6         [2576]
640 IF sprung=1 THEN GOTO 660              [1521]
650 CLS #5:LOCATE #5,7,1:INPUT #5,"Eingabe [4535]
 des Datum's                ",datum$
660 LOCATE #6,6,1:PRINT #6,datum$          [482]
670 '                                      [117]
680 ' sammelbestellerdaten laden           [954]
690 '                                      [117]
700 CLS #5:LOCATE #5,7,1:INPUT #5,"Sammelb [6823]
estellerdaten laden  ?    J/N      ",j$
710 IF j$ ="j" OR j$="J" THEN 720 ELSE IF  [2807]
sprung=1 THEN GOTO 900
720 CLS#1:LOCATE #1,17,1:PRINT #1,"Sammelb [3001]
esteller - Daten "
730 CLS #3:LOCATE #3,13,6:PRINT #3,"Bitte  [5462]
die Datendiskette einlegen,"
740 LOCATE #3,16,8:PRINT #3,"dann eine Tas [3404]
te dr}cken."
750 CALL &BB06                             [393]
760 CLS #3:LOCATE #3,12,6:PRINT #3,"Daten  [2942]
werden von Diskette geladen."
770 OPENIN "QSAMMEL.DAT"                   [2050]
780 INPUT #9,datuml$                       [1934]
790 INPUT #9,allkonto                      [1489]
800 INPUT #9,allstueck                     [1306]
810 INPUT #9,allpreis                      [1555]
820 INPUT #9,allretur                      [883]
830 INPUT #9,allretursumme                 [1263]
840 INPUT #9,allreturzins                  [1525]
850 INPUT #9,allgewinn                     [985]
860 CLOSEIN                                [752]
870 '                                      [117]
880 ' anzeigen der sammelbestellerdaten    [2667]
890 '                                      [117]
900 CLS#3:CLS#1:LOCATE #1,7,1:PRINT #1,"Sa [5572]
mmelbesteller - Daten     vom   ";datuml$

910 LOCATE #3,7,1:PRINT #3,"Anzahl aller b [3609]
estellten Artikel              ="
```

Listing SAMMEL

```
920 LOCATE #3,7,3:PRINT #3,"Summe   aller b [5605]
estellten Artikel              ="
930 LOCATE #3,7,5:PRINT #3,"Anzahl aller R [3667]
etouren                        ="
940 LOCATE #3,7,7:PRINT #3,"Summe   aller R [4010]
etouren                        ="
950 LOCATE #3,7,9:PRINT #3,"Summe   aller R [3395]
etourzinsen                    ="
960 LOCATE #3,7,11:PRINT #3,"Gewinnbeteili [3423]
gung                           ="
970 LOCATE #4,5,1:PRINT #4,USING"######";a [2653]
llstueck
980 LOCATE #4,5,3:PRINT #4,USING"######.## [4080]
 DM";allpreis
990 LOCATE #4,5,5:PRINT #4,USING"######";a [3257]
llretur
1000 LOCATE #4,5,7:PRINT #4,USING"######.# [1748]
# DM";allretursumme
1010 LOCATE #4,5,9:PRINT #4,USING"######.# [3311]
# DM";allreturzins
1020 LOCATE #4,5,11:PRINT #4,USING"######. [1821]
## DM";allgewinn
1030 LOCATE #2,6,1:PRINT #2,USING "######. [1784]
## DM";allkonto
1040 '                                     [117]
1050 ' liste der mitbesteller              [1794]
1060 '                                     [117]
1070 CLS #5:LOCATE #5,7,1:INPUT #5,"Liste [5876]
der Mitbesteller          J/N      ",j$
1080 IF j$ ="j" OR j$="J" THEN 1090 ELSE 4 [1697]
200
1090 CLS #1:CLS #2:CLS #3:CLS #4           [1769]
1100 LOCATE #3,3,2:PRINT #3,"-QA- Christel [1552]
     Kippenberg "
1110 LOCATE #3,3,3:PRINT #3,"-QB- Katrin   [2541]
     Kippenberg "
1120 LOCATE #3,3,4:PRINT #3,"-QC- J¦rg     [3315]
     Kippenberg "
1130 LOCATE #3,3,5:PRINT #3,"-QD- Ingo     [3982]
     Kippenberg "
1140 LOCATE #3,3,6:PRINT #3,"-QE- Marieche [3126]
n    Lorat "
1150 LOCATE #3,3,7:PRINT #3,"-QF- Ralf     [4590]
     Lorat "
1160 LOCATE #3,3,8:PRINT #3,"-QG- Elke     [1846]
     Lorat "
1170 LOCATE #3,3,9:PRINT #3,"-QH- Marita   [3051]
     Jones "
1180 LOCATE #3,3,10:PRINT #3,"-QI- Erna    [2228]
     Klusmeyer "
1190 LOCATE #3,3,11:PRINT #3,"-QJ- Ulrike  [3473]
     Klusmeyer "
1200 LOCATE #3,33,2:PRINT #3,"-QK- Margret [2732]
     Stelling "
1210 LOCATE #3,33,3:PRINT #3,"-QL- Waltrau [3734]
d    Duensing "
1220 LOCATE #3,33,4:PRINT #3,"-QM- Corneli [3443]
a    Heidrich "
1230 LOCATE #3,33,5:PRINT #3,"-QN- Magitta [3561]
     Hartmann "
1240 '                                     [117]
1250 '                                     [117]
1260 '                                     [117]
1270 '                                     [117]
1280 CLS #5:LOCATE #5,7,1:INPUT #5,"Welche [5700]
r Mitbesteller  ?                  ",mitbes
t$
1290 mitbest$=UPPER$(mitbest$)             [2639]
1300 :LOCATE #1,3,1:PRINT #1,"-";mitbest$; [3288]
"-"
1310 '                                     [117]
1320 ' mitbestellerdaten laden             [2604]
1330 '                                     [117]
1340 CLS #5:LOCATE #5,7,1:INPUT #5,"Mitbes [4687]
tellerdaten laden  ?       J/N      ",j$
1350 IF j$ ="j" OR j$="J" THEN 1360 ELSE 1 [1175]
960
1360 CLS #3:LOCATE #3,13,6:PRINT #3,"Bitte [5462]
 die Datendiskette einlegen,"
1370 LOCATE #3,16,8:PRINT #3,"dann eine Ta [3404]
ste dr}cken."
1380 CALL &BB06                            [393]
1390 CLS #3:LOCATE #3,12,6:PRINT #3,"Daten [2942]
 werden von Diskette geladen."
1400 OPENIN mitbest$+".dat"                [1886]
1410 INPUT #9,name1$                       [1454]
1420 INPUT #9,name2$                       [1468]
1430 INPUT #9,stueck                       [949]
```

Listing SAMMEL

```
1440 INPUT #9,gesstueck                    [1014]
1450 INPUT #9,retur                        [1181]
1460 INPUT #9,gesretur                     [1213]
1470 INPUT #9,retursumme                   [1769]
1480 INPUT #9,gesretursumme                [1612]
1490 INPUT #9,preis                        [1543]
1500 INPUT #9,gespreis                     [935]
1510 INPUT #9,raten                        [954]
1520 INPUT #9,bezraten                     [1772]
1530 INPUT #9,monat$                       [1059]
1540 INPUT #9,ratpreis                     [996]
1550 INPUT #9,datum1$                      [1934]
1560 INPUT #9,konto                        [941]
1570 CLOSEIN                               [752]
1580 CLS #3:jump=1:GOSUB 1630              [997]
1590 GOTO 1960                             [339]
1600 '                                     [117]
1610 ' mitbestellerdaten anzeigen          [2067]
1620 '                                     [117]
1630 CLS #3                                [371]
1640 CLS #1:LOCATE #1,3,1:PRINT #1,"-";mit [1887]
best$;"-"
1650 LOCATE #1,10,1:PRINT #1,name1$;" "nam [2449]
e2$
1660 LOCATE #1,39,1:PRINT #1," vom ";datum [1800]
1$
1670 LOCATE #3,7,1:PRINT #3,"Anzahl der be [4387]
stellten Artikel               ="
1680 LOCATE #3,7,2:PRINT #3,"Summe  der be [3571]
stellten Artikel               ="
1690 LOCATE #3,7,3:PRINT #3,"Anzahl der Re [4532]
touren                         ="
1700 LOCATE #3,7,4:PRINT #3,"Summe  der Re [4839]
touren                         ="
1710 LOCATE #3,7,5:PRINT #3,"Anzahl der Ra [4422]
ten                            ="
1720 LOCATE #3,7,6:PRINT #3,"Preis  der Ra [3276]
ten                            ="
1730 LOCATE #3,7,7:PRINT #3,"F{lligkeit de [3354]
r 1. Rate                      ="
1740 LOCATE #3,7,8:PRINT #3,"Bezahlte Rate [2765]
n                              ="
1750 LOCATE #3,7,10:PRINT #3,"Anzahl aller [3169]
 bestellten Artikel            ="
1760 LOCATE #3,7,11:PRINT #3,"Summe  aller [4296]
 bestellten Artikel            ="
1770 LOCATE #3,7,12:PRINT #3,"Anzahl aller [4353]
 Retouren                      ="
1780 LOCATE #3,7,13:PRINT #3,"Summe  aller [3203]
 Retouren                      ="
1790 CLS#2:LOCATE #2,6,1:PRINT #2,USING "# [3274]
####.## DM";konto
1800 CLS #4:LOCATE #4,5,1:PRINT #4,USING"# [4619]
#####";stueck
1810 LOCATE #4,5,2:PRINT #4,USING"######.# [3164]
# DM";preis
1820 LOCATE #4,5,3:PRINT #4,USING"######"; [2487]
retur
1830 LOCATE #4,5,4:PRINT #4,USING"######.# [3614]
# DM";retursumme
1840 LOCATE #4,5,5:PRINT #4,USING"######"; [3241]
raten
1850 LOCATE #4,5,6:PRINT #4,USING"######.# [3006]
# DM";ratpreis
1860 LOCATE #4,8,7:PRINT #4,monat$         [1770]
1870 LOCATE #4,5,8:PRINT #4,USING"######"; [2446]
bezraten
1880 LOCATE #4,5,10:PRINT #4,USING"######" [2293]
;gesstueck
1890 LOCATE #4,5,11:PRINT #4,USING"######. [1928]
## DM";gespreis
1900 LOCATE #4,5,12:PRINT #4,USING"######" [2658]
;gesretur
1910 LOCATE #4,5,13:PRINT #4,USING"######. [5095]
## DM";gesretursumme
1920 RETURN                                [555]
1930 '                                     [117]
1940 ' neue bestellung eingeben            [1216]
1950 '                                     [117]
1960 CLS #5:LOCATE #5,7,1:INPUT #5,"Neue B [6371]
estellung eingeben ?      J/N      ",j$
1970 IF j$ ="j" OR j$="J" THEN 1980 ELSE 2 [1638]
290
1980 IF bezraten<raten THEN GOTO 1990 ELSE [3068]
 2060
1990 CLS #3:LOCATE #3,22,6:PRINT#3,"A C H  [2954]
T U N G !!!"
```

Listing SAMMEL

```
2000 LOCATE #3,11,8:PRINT#3,"Es sind noch  [5246]
nicht alle Raten bezahlt !"
2010 LOCATE #3,8,10:PRINT#3,"Neue Bestellu [6445]
ngen sind nur nach Eingabe einer"
2020 LOCATE #3,9,11:PRINT#3,"neuen Datei,  [7566]
zum Beispiel - ";mitbest$"1 - m|glich."
2030 CLS #5:LOCATE #5,7,1:INPUT #5,"Neue D [4997]
atei erstellen  ?          J/N     ",j$
2040 IF j$ ="j" OR j$="J" THEN 2050 ELSE 2 [1085]
540
2050 CLS#3:RUN 610                          [589]
2060 CLS #5:LOCATE #5,7,1:INPUT #5,"Vornam [6528]
e des Mitbestellers   ",name1$
2070 CLS #5:LOCATE #5,7,1:INPUT #5,"Nachna [4532]
me des Mitbestellers   ",name2$
2080 LOCATE #1,10,1:PRINT #1,name1$;" "nam [2449]
e2$
2090 LOCATE #1,39,1:PRINT #1,datum1$       [1539]
2100 stueck=0:retur=0:preis=0:raten=0:ratp [4793]
reis=0:monat=0:bezraten=0
2110 CLS #5:LOCATE #5,7,1:INPUT #5,"Anzahl [5897]
 der bestellten Artikel ?            ",stueck
2120 gesstueck=gesstueck+stueck            [1844]
2130 allstueck=allstueck+stueck            [1739]
2140 CLS #5:LOCATE #5,7,1:INPUT #5,"Summe  [5055]
der bestellten Artikel ?             ",preis
2150 gespreis=gespreis+preis               [1751]
2160 konto=konto-preis                     [898]
2170 allkonto=allkonto-preis               [1487]
2180 allpreis=allpreis+preis               [1208]
2190 gewpreis=gewpreis+preis               [1982]
2200 gewinn=gewpreis/100*5                 [1264]
2210 allgewinn=allgewinn+gewinn            [2466]
2220 CLS #5:LOCATE #5,7,1:INPUT #5,"Ratenz [4913]
ahlung                     J/N     ",j$
2230 IF j$ ="j" OR j$="J" THEN 2240 ELSE 2 [1460]
290
2240 CLS #5:LOCATE #5,7,1:INPUT #5,"Anzahl [4564]
 der Raten  ?                        ",raten
2250 CLS #5:LOCATE #5,7,1:INPUT #5,"F{llig [4292]
keit der 1. Rate                     ",monat$

2260 '                                      [117]
2270 'retouren eingeben                    [1292]
2280 '                                      [117]
2290 CLS #5:LOCATE #5,7,1:INPUT #5,"Retour [5195]
en eingeben            J/N     ",j$
2300 IF j$ ="j" OR j$="J" THEN 2310 ELSE 2 [1588]
400
2310 CLS #5:LOCATE #5,7,1:INPUT #5,"Anzahl [4538]
 der Retouren                        ",retur

2320 gesretur=gesretur+retur               [2751]
2330 allretur=allretur+retur               [2143]
2340 CLS #5:LOCATE #5,7,1:INPUT #5,"Summe  [6063]
der Retouren                         ",retursu
mme
2350 allkonto=allkonto+retursumme          [1577]
2360 konto=konto+retursumme                [1721]
2370 gesretursumme=gesretursumme+retursumm [3777]
e
2380 allretursumme=allretursumme+retursumm [2443]
e
2390 '                                      [117]
2400 IF jump=1 THEN GOTO 2450              [1028]
2410 ratpreis=(preis-retursumme)/raten     [1578]
2420 '                                      [117]
2430 'gutschrift von retourzinsen          [2590]
2440 '                                      [117]
2450 CLS #5:LOCATE #5,7,1:INPUT #5,"Gutsch [5323]
rift von Retourzinsen ?  J/N     ",j$
2460 IF j$ ="j" OR j$="J" THEN 2470 ELSE 2 [1827]
540
2470 CLS #5:LOCATE #5,7,1:INPUT #5,"Summe  [7332]
der Retourzinsen
                                     ",returzi
ns
2480 allkonto=allkonto+returzins           [894]
2490 konto=konto+returzins                 [1830]
2500 allreturzins=allreturzins+returzins   [3768]

2510 '                                      [117]
2520 ' einzelbetr{ge einzahlen             [1821]
2530 '                                      [117]
2540 IF raten>0 OR raten>bezraten THEN GOT [1733]
O 2630
2550 CLS #5:LOCATE #5,7,1:INPUT #5,"Einzel [4581]
betr{ge einzahlen  ?      J/N     ",j$
```

Listing SAMMEL

```
2560 IF j$ ="j" OR j$="J" THEN 2570 ELSE 2 [1285]
630
2570 CLS #5:LOCATE #5,7,1:INPUT #5,"Welche [4786]
n Betrag einzahlen  ? ",betrag
2580 konto=konto+betrag                    [1253]
2590 allkonto=allkonto+betrag:GOTO 2700    [1441]
2600 '                                      [117]
2610 ' ratenbetr{ge einzahlen              [2106]
2620 '                                      [117]
2630 IF raten=0 OR raten=bezraten THEN GOT [2758]
O 2740
2640 CLS #5:LOCATE #5,7,1:INPUT #5,"Ratenb [6623]
etr{ge einzahlen  ?        J/N     ",j$
2650 IF j$ ="j" OR j$="J" THEN 2660 ELSE 2 [1476]
740
2660 CLS #5:LOCATE #5,7,1:INPUT #5,"Welche [6628]
n Ratenbetrag einzahlen  ? ",ratenbetrag
2670 konto=konto+ratenbetrag               [1387]
2680 allkonto=allkonto+ratenbetrag         [901]
2690 bezraten=bezraten+1                   [928]
2700 IF konto >=0 THEN  bezraten=raten     [3084]
2710 '                                      [117]
2720 ' mitbestellerdaten anzeigen          [2067]
2730 '                                      [117]
2740 CLS #5:LOCATE #5,7,1:INPUT #5,"Mitbes [5372]
tellerdaten anzeigen  ?   J/N     ",j$
2750 datum1$=datum$                        [1045]
2760 IF j$ ="j" OR j$="J" THEN GOSUB 1630  [2458]
ELSE 2830
2770 '                                      [117]
2780 ' ***** alle daten abspeichern *****  [2534]
2790 '                                      [117]
2800 '                                      [117]
2810 ' mitbestellerdaten abspeichern       [2337]
2820 '                                      [117]
2830 CLS #5:LOCATE #5,7,1:INPUT #5,"Mitbes [6294]
tellerdaten abspeichern  ? J/N     ",j$
2840 IF j$ ="j" OR j$="J" THEN 2860 ELSE 3 [1912]
240
2850 '                                      [117]
2860 CLS #3:LOCATE #3,13,6:PRINT #3,"Bitte [5462]
 die Datendiskette einlegen,"
2870 LOCATE #3,16,8:PRINT #3,"dann eine Ta [3404]
ste dr}cken."
2880 datum1$=datum$:CALL &BB06             [1724]
2890 CLS #3:LOCATE #3,12,6:PRINT #3,"Daten [3550]
 werden auf Diskette gespeichert"
2900 OPENOUT mitbest$+".dat"               [1201]
2910 PRINT #9,name1$                       [1600]
2920 PRINT #9,name2$                       [1614]
2930 PRINT #9,stueck                       [1508]
2940 PRINT #9,gesstueck                    [1388]
2950 PRINT #9,retur                        [1460]
2960 PRINT #9,gesretur                     [1289]
2970 PRINT #9,retursumme                   [2022]
2980 PRINT #9,gesretursumme                [1863]
2990 PRINT #9,preis                        [1294]
3000 PRINT #9,gespreis                     [1124]
3010 PRINT #9,raten                        [1448]
3020 PRINT #9,bezraten                     [1824]
3030 PRINT #9,monat$                       [1350]
3040 PRINT #9,ratpreis                     [1817]
3050 PRINT #9,datum1$                      [1306]
3060 PRINT #9,konto                        [1430]
3070 CLOSEOUT                              [902]
3080 '                                      [117]
3090 ' sammelbestellerdaten abspeichern    [1345]
3100 '                                      [117]
3110 OPENOUT "QSAMMEL.DAT"                 [1567]
3120 PRINT #9,datum1$                      [1306]
3130 PRINT #9,allkonto                     [986]
3140 PRINT #9,allstueck                    [925]
3150 PRINT #9,allpreis                     [1595]
3160 PRINT #9,allretur                     [907]
3170 PRINT #9,allretursumme                [1175]
3180 PRINT #9,allreturzins                 [1147]
3190 PRINT #9,allgewinn                    [1532]
3200 CLOSEOUT:CLS#3                        [1578]
3210 '                                      [117]
3220 ' daten ausdrucken                    [1847]
3230 '                                      [117]
3240 CLS #5:LOCATE #5,7,1:INPUT #5,"Daten  [3469]
ausdrucken            ? J/N     ",j$
3250 IF j$ ="j" OR j$="J" THEN 3290 ELSE 4 [1417]
200
3260 '                                      [117]
3270 ' alle daten ausdrucken               [1904]
```

Listing SAMMEL

```
3280 '                                  [117]
3290 CLS #5:LOCATE #5,7,1:INPUT #5,"Alle D [4060]
aten ausdrucken         ? J/N    ",j$
3300 IF j$ ="j" OR j$="J" THEN 3310 ELSE 3 [1641]
700
3310 CLS#3:LOCATE #3,7,6:PRINT#3,"Bitte Dr [5917]
ucker einschalten, und Papier einlegen"

3320 LOCATE #3,17,8:PRINT#3,"dann eine Tas [3437]
te dr}cken"
3330 CALL &BB06                         [393]
3340 PRINT#8                            [343]
3350 CLS#3:LOCATE #3,12,6:PRINT#3,"Alle Da [3053]
ten werden ausgedruckt"
3360 PRINT#8,CHR$(14);CHR$(27);"-";CHR$(1) [6058]
;" S A M M E L B E S T E L L U N G";CHR$(2
7);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(10);
CHR$(10)
3370 PRINT#8,CHR$(9);CHR$(14);CHR$(27);"-" [8684]
;CHR$(1);" Q U E L L E  V E R S A N D";CHR
$(27);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(1
0);CHR$(10)
3380 PRINT#8,CHR$(9);name1$;" ";name2$;CHR [6146]
$(9);CHR$(9);datum$
3390 PRINT#8                            [343]
3400 PRINT#8                            [343]
3410 PRINT#8,"Kontostand des Sammelbestell [6651]
ers          =",USING"######.## DM";allk
onto
3420 PRINT#8                            [343]
3430 PRINT#8,"Anzahl aller bestellten Arti [5720]
kel          =",USING"######";allstueck

3440 PRINT#8,"Summe  aller bestellten Arti [6036]
kel          =",USING"######.## DM";allp
reis
3450 PRINT#8,"Anzahl aller Retouren        [6272]
             =",USING"######";allretur

3460 PRINT#8,"Summe  aller Retouren        [4348]
             =",USING"######.##";allretu
rsumme
3470 PRINT#8,"Summe  aller Retourzinsen    [4577]
             =",USING"######.## DM";allr
eturzins
3480 PRINT#8,"Gewinnbeteiligung            [4815]
             =",USING"######.## DM";allg
ewinn
3490 PRINT#8                            [343]
3500 PRINT#8,"Kontostand des Mitbestellers [5254]
             =",USING"######.## DM";kont
o
3510 PRINT#8                            [343]
3520 PRINT#8,"Anzahl der bestellten Artike [7215]
l            =",USING"######";stueck
3530 PRINT#8,"Summe  der bestellten Artike [6487]
l            =",USING"######.## DM";prei
s
3540 PRINT#8,"Anzahl der Retouren          [6065]
             =",USING"######";retur
3550 PRINT#8,"Summe  der Retouren          [4494]
             =",USING"######.##";retursu
mme
3560 PRINT#8,"Anzahl der Raten             [5370]
             =",USING"######";raten
3570 PRINT#8,"Preis  der Raten             [4136]
             =",USING"######.## DM";ratp
reis
3580 PRINT#8,"F{lligkeit der 1. Rate       [4088]
             =           ";monat$
3590 PRINT#8,"Bezahlte Raten               [4306]
             =",USING"######";bezraten
3600 PRINT#8                            [343]
3610 PRINT#8,"Anzahl aller bestellten Arti [6042]
kel          =",USING"######";gesstueck

3620 PRINT#8,"Summe  aller bestellten Arti [6439]
kel          =",USING"######.## DM";gesp
reis
3630 PRINT#8,"Anzahl aller Retouren        [6271]
             =",USING"######";gesretur

3640 PRINT#8,"Summe  aller Retouren        [4276]
             =",USING"######.##";gesretu
rsumme
3650 PRINT#8                            [343]
3660 PRINT#8,"============================ [4549]
======================================="
```

Listing SAMMEL

```
3670 '                                  [117]
3680 ' sammelbestellerdaten ausdrucken  [1417]
3690 '                                  [117]
3700 CLS #5:LOCATE #5,7,1:INPUT #5,"Sammel [6829]
bestellerdaten ausdrucken  ? J/N ",j$
3710 IF j$ ="j" OR j$="J" THEN 3720 ELSE 3 [1358]
940
3720 CLS#3:LOCATE #3,7,6:PRINT#3,"Bitte Dr [5917]
ucker einschalten, und Papier einlegen"

3730 LOCATE #3,17,8:PRINT#3,"dann eine Tas [3437]
te dr}cken"
3740 CALL &BB06                         [393]
3750 PRINT#8                            [343]
3760 CLS#3:LOCATE #3,10,6:PRINT#3,"Sammelb [6390]
estellerdaten werden ausgedruckt"
3770 PRINT#8,CHR$(14);CHR$(27);"-";CHR$(1) [6058]
;" S A M M E L B E S T E L L U N G";CHR$(2
7);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(10);
CHR$(10)
3780 PRINT#8,CHR$(9);CHR$(14);CHR$(27);"-" [7905]
;CHR$(1);"Q U E L L E  V E R S A N D";CHR$
(27);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(10
);CHR$(10)
3790 PRINT#8                            [343]
3800 PRINT#8                            [343]
3810 PRINT#8,"Kontostand des Sammelbestell [6651]
ers          =",USING"######.## DM";allk
onto
3820 PRINT#8                            [343]
3830 PRINT#8,"Anzahl aller bestellten Arti [5720]
kel          =",USING"######";allstueck

3840 PRINT#8,"Summe  aller bestellten Arti [6036]
kel          =",USING"######.## DM";allp
reis
3850 PRINT#8,"Anzahl aller Retouren        [6272]
             =",USING"######";allretur

3860 PRINT#8,"Summe  aller Retouren        [4348]
             =",USING"######.##";allretu
rsumme
3870 PRINT#8,"Summe  aller Retourzinsen    [4644]
             =",USING"######.##";allretu
rzins
3880 PRINT#8,"Gewinnbeteiligung            [4815]
             =",USING"######.## DM";allg
ewinn
3890 PRINT#8                            [343]
3900 PRINT#8,"============================ [4549]
======================================="
3910 '                                  [117]
3920 ' mitbestellerdaten ausdrucken     [2307]
3930 '                                  [117]
3940 CLS #5:LOCATE #5,7,1:INPUT #5,"Mitbes [5068]
tellerdaten ausdrucken    ? J/N    ",j$
3950 IF j$ ="j" OR j$="J" THEN 3960 ELSE 4 [1517]
200
3960 CLS#3:LOCATE #3,7,6:PRINT#3,"Bitte Dr [5917]
ucker einschalten, und Papier einlegen"
3970 LOCATE #3,17,8:PRINT#3,"dann eine Tas [3437]
te dr}cken"
3980 CALL &BB06                         [393]
3990 PRINT#8                            [343]
4000 CLS#3:LOCATE #3,12,6:PRINT#3,"Mitbest [4214]
ellerdaten werden ausgedruckt"
4010 PRINT#8,CHR$(14);CHR$(27);"-";CHR$(1) [6058]
;" S A M M E L B E S T E L L U N G";CHR$(2
7);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(10);
CHR$(10)
4020 PRINT#8,CHR$(9);CHR$(14);CHR$(27);"-" [7905]
;CHR$(1);"Q U E L L E  V E R S A N D";CHR$
(27);"-";CHR$(0);CHR$(10);CHR$(20);CHR$(10
);CHR$(10)
4030 PRINT#8,CHR$(9);name1$;" ";name2$;CHR [6146]
$(9);CHR$(9);datum$
4040 PRINT#8                            [343]
4050 PRINT#8                            [343]
4060 PRINT#8,"Kontostand des Mitbestellers [5254]
             =",USING"######.## DM";kont
o
4070 PRINT#8                            [343]
4080 PRINT#8,"Anzahl der bestellten Artike [7215]
l            =",USING"######";stueck
4090 PRINT#8,"Summe  der bestellten Artike [6487]
l            =",USING"######.## DM";prei
s
4100 PRINT#8,"Anzahl der Retouren          [6065]
             =",USING"######";retur
```

Listing SAMMEL

```
4110 PRINT#8,"Anzahl der Raten                     [5370]
              =",USING"######";raten
4120 PRINT#8,"Preis  der Raten                     [4136]
              =",USING"######.## DM";ratp
reis
4130 PRINT#8,"F{lligkeit der 1. Rate               [4332]
              =               ";monat$
4140 PRINT#8,"Bezahlte Raten                       [4306]
              =",USING"######";bezraten

4150 PRINT#8                                       [343]
4160 PRINT#8,"============================ [4549]
========================================"
4170 '                                             [117]
4180 ' disketteninhalt anzeigen                    [1583]
4190 '                                             [117]
4200 CLS #5:LOCATE #5,7,1:INPUT #5,"Disket [4705]
teninhalt anzeigen          J/N   ",j$
4210 IF j$ ="j" OR j$="J" THEN 4220 ELSE 4 [1719]
260
4220 MODE 2:¦DIR:PRINT "Bitte eine Taste d [6210]
r}cken.":CALL &BB06:sprung=1:GOTO 290
4230 '                                             [117]
4240 ' programm beenden                            [1545]
4250 '                                             [117]
4260 CLS #3:LOCATE #3,10,6:PRINT#3,"P R O  [3397]
G R A M M    B E E N D E N   ????"
4270 :LOCATE #3,4,10:PRINT#3,"ACHTUNG !!!  [5795]
Sind alle neuen Daten auch gespeichert ?"

4280 CLS #5:LOCATE #5,7,1:INPUT #5,"Sind S [4550]
ie sicher ?                 J/N   ",j$
4290 IF j$ ="j" OR j$="J" THEN 4300 ELSE C [2622]
LS#3:sprung=1:GOTO 290
4300 CLG:END:GOTO 10                               [1459]
4310 '                                             [117]
4320 ' sammelbestellerdatei erstellen              [2028]
4330 '                                             [117]
4340 CLS #3:LOCATE #3,6,4:PRINT#3,"Soll di [6238]
e Sammelbestellerdatei erstellt werden ? "

4350 CLS #5:LOCATE #5,7,1:INPUT #5,"Sind S [4550]
ie sicher ?                 J/N   ",j$
4360 IF j$ ="j" OR j$="J" THEN 4370 ELSE R [2298]
ETURN
4370 allkonto=0:allstueck=0:allpreis=0:all [6515]
retur=0:allretursumme=0:allreturzins=0:gew
inn=0
4380 CLS#5:CLS#1:LOCATE #1,17,1:PRINT #1," [4118]
Sammelbesteller - Datei "
4390 CLS #3:LOCATE #3,13,6:PRINT #3,"Bitte [5462]
 die Datendiskette einlegen,"
4400 LOCATE #3,16,8:PRINT #3,"dann eine Ta [3404]
ste dr}cken."
4410 CALL &BB06                                    [393]
4420 CLS #3:LOCATE #3,12,6:PRINT #3,"Samme [4388]
lbestellerdatei wird er¦ffnet. "
4430 OPENOUT "QSAMMEL.DAT"                         [1567]
4440 PRINT #9,datum1$                              [1306]
4450 PRINT #9,allkonto                             [986]
4460 PRINT #9,allstueck                            [925]
4470 PRINT #9,allpreis                             [1595]
4480 PRINT #9,allretur                             [907]
4490 PRINT #9,allretursumme                        [1175]
4500 PRINT #9,allreturzins                         [1147]
4510 PRINT #9,allgewinn                            [1532]
4520 CLOSEOUT                                      [902]
4530 CLS #5:CLS #3:LOCATE #3,20,3:PRINT #3 [3269]
," A C H T U N G ! ! !"
4540 LOCATE #3,20,4:PRINT #3," =========== [2843]
========"
4550 LOCATE #3,5,6:PRINT #3,"Die Sammelbes [7112]
tellerdatei ist erstellt.Sie sollten "

4560 LOCATE #3,3,8:PRINT #3,"in der Sprung [6160]
adresse  610  das REM-Zeichen einsetzen."

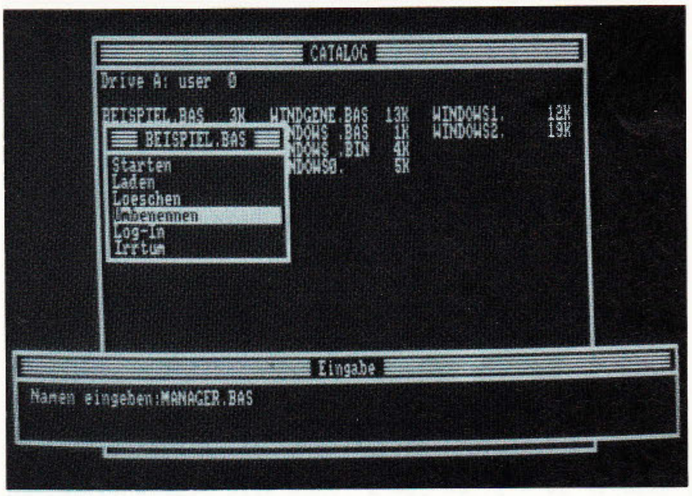
4570 LOCATE #3,15,11:PRINT #3,"Bitte eine  [3775]
Taste dr}cken."
4580 CALL &BB06                                    [393]
4590 RETURN                                        [555]
```

Listing SAMMEL

Windows? Null Problemo!

Windows einfach gehandhabt

Welcher Computer-Neuling auf dem CPC kennt nicht das Problem mit den Windows? Window Manager ist ein Programm, das all denen, die dieses Problem haben, helfen kann.



Bilder lassen sich auch ohne Probleme behandeln

Dieses Programm wurde konzipiert, um den Zusatzspeicher des CPC 6128, im folgenden nur noch als "Stack" bezeichnet, nutzbar zu machen. In erster Linie handelt es sich um Befehle zur Window-Verwaltung, die in der Art von Pull-down-Menüs benutzt werden können. Dazu wurde ein neues Koordinatensystem festgelegt: der Ursprung befindet sich in der oberen linken Ecke des Bildschirms, dieser Punkt heißt demnach (0,0). Davon ausgehend befindet sich der entfernteste Punkt in der unteren rechten Ecke und heißt (79,199). Alle Befehle, die den Bildschirm verändern (mit Ausnahme des CLOSE- und SHOW-Befehls), setzen voraus, daß in Bildschirmmodus 2 gearbeitet wird. Das heißt aber nicht, daß die Befehle in einem Grafik-Programm, das in einem anderen Modus arbeitet, nicht zu gebrauchen wären, ganz im Gegenteil! Das Programm benötigt nur sechs Kilobyte des Hauptspeichers, sämtliche Bilddaten werden in Stack gespeichert. Die oberste Speicherstelle des Programms liegt unter &A000, daher haben Sie noch Platz für andere M-Codes. Damit auch Benutzer des 464 und des 664 in den Genuß dieser Befehlserweiterung kommen können, wurde die Möglichkeit geschaffen, die Bilddaten auch im normalen Speicher abzulegen. Die dk'tronics-Erweiterung kann jedoch genutzt werden. (Das Programm wurde auf einem CPC 664 mit dieser Erweiterung geschrieben).

Für die folgende Befehlsbeschreibung werden einige Definitionen benötigt. Es gilt:

w,x,y,z: Eine Zahl oder eine einfache Zahlen-Variable.

vx,vy,vz: Eine Variable vom Typ Integer, die als solche zu Beginn des Basic-Programms durch "DEFINT" definiert werden muß oder in Verbindung mit dem %-Zeichen benutzt wird.

Wichtig: Dieser Zahl muß zuvor irgendein Wert zugeordnet werden und im RSX-Befehl ein "Klammeraffe" (@) vorangestellt werden.

s: Ein String oder eine String-Variable. Läuft die Befehlserweiterung auf einem CPC 464, so muß eine String-Variable definiert werden und dieser ebenfalls der "Klammeraffe" vorangestellt werden. Im folgenden werden die Befehle ohne RSX-Strich angeführt. Wird eine andere Anzahl von Parametern verwendet als erlaubt, so wird der Befehl ignoriert, eine Fehlermeldung erscheint nicht.

Befehle zur Window-Verwaltung

OPEN,w,x1,y1,x2,y2
Dieser Befehl definiert ein Fenster mit der Nummer w. Man beachte, daß bis zu 255 verschiedene Fenster definiert werden können, sofern der Speicherplatz ausreicht. Man sollte jedoch nur Nummern von 1 bis 199 benutzen, da der MENÜ-Befehl die Werte 200-255 belegt. x1 und y1 geben die Koordinaten für die obere linke Ecke des Fensters an, x2 legt die Ausdehnung des Fensters in Byte in der Horizontalen und y2 die Ausdehnung in Byte in der Vertikalen fest.

Mögliche Werte:
x1:0-79;
y1:0-199;
x2:1-80;
y2:1-200.

Der Befehl hat unter Normalbedingungen (siehe Befehl ART) keine sichtbaren Auswirkungen auf dem Bildschirm, jedoch wird der Ausschnitt in den Stack kopiert, die Verwaltung dieser Ausschnitte übernimmt ein internes System, Sie brauchen sich also nicht um Speicherplatzverbrauch oder dergleichen zu kümmern. Sollte schon ein Fenster mit der Nummer w. existieren, so wird dies erst geschlossen und dann der Befehl ausgeführt.
Beispiel: **OPEN,1,3,24,20,80**

CLOSE,w
Der Sinn der OPEN-CLOSE-Kombination ist es, Pulldown-Strukturen erzeugen zu können. w. gibt die Nummer eines bereits geöffneten Fensters an. Infolge dieses Befehls wird der ursprüngliche Bildschirminhalt in dem definierten Bereich wieder hergestellt, das heißt, mögliche Veränderungen in dem Fenster werden rückgängig gemacht. Als weitere Folge wird der Speicherplatz für diesen Ausschnitt im Stack wieder freigegeben, das heißt, nach diesem Befehl existiert das Window w. als Definition nicht mehr.
Beispiel: **CLOSE,1**

CLOSE (ohne Parameter)
Sämtliche Fenster und Menüfenster werden geschlossen, das heißt, ihr ursprünglicher Bildschirminhalt erscheint wieder. Dabei ist auf die Reihenfolge zu achten: zuerst werden alle Menü-Fenster in der umgekehrten Reihenfolge ihres Erscheinens und dann alle Fenster von 199 bis 1 rückwärts geschlossen. Daher ist es am günstigsten, mit einer niedrigen Nummer beim Öffnen zu beginnen und daraufhin immer nur höhere Fensternummern zu benutzen.

CLEAR,w.
Das Fenster w. wird aus dem Stack gelöscht, ohne jedoch vorher den entsprechenden Bildschirmbereich zu überdecken (im Gegensatz zum CLOSE-Kommando).

CLEAR (ohne Parameter)
Der gesamte Stack wird gelöscht, das heißt, sämtliche OPEN- und MENÜ-Definitionen gelten nicht mehr.

l,SHOW,w
Hat den gleichen Effekt wie das CLOSE-Kommando, jedoch bleiben die Bild-

schirmdaten im Stack erhalten, das Window wird also nicht "geschlossen".

SHOW,w,x,y
Bietet die Möglichkeit, den durch das OPEN-Kommando abgespeicherten Bildschirmbereich an einer beliebigen Stelle auf dem Bildschirm darzustellen. Mit Hilfe einer OPEN/SHOW/CLEAR-Befehlsfolge lassen sich so sehr schnelle und einfach zu programmierende Verschieberoutinen, z.B. für ein Grafikprogramm, realisieren.
Beispiel: **SHOW,1,10,90**

NEWPOS,w,x,y
Der Kopie des ursprünglichen Bildschirmausschnittes eines Fensters w. wird eine neue Ursprungskoordinate zugewiesen. Dies hat natürlich nur Sinn, wenn dies zum Zweck einer Verschieberoutine, wie beim SHOW-Kommando beschrieben, erfolgt. Dementsprechend kann die Befehlsfolge
OPEN,w,x1,y1,x2,y2: SHOW,w,x3,y3:CLEAR,w
ersetzt werden durch
OPEN,w,x1,y1,x2,y2: NEWPOS,w,x3,y3:CLOSE,w.
Die Auswirkungen sind die gleichen.

GETLOC,w,vx1,vy1,vx2,vy2
Den letzten vier angegebenen Variablen (mit vorangestelltem 'Klammeraffen') werden die Ausmaße des Fensters w übertragen, gemäß der Definition mit OPEN oder der Änderung durch NEWPOS.
Beispiel: **a%=1:b%=1:c%=1:d%=1: GETLOC,1, a%, b%,c%, d%: PRINT a%;b%;c%;d%**

CLS,w
Der definierte Bereich des Fensters w. auf dem Bildschirm wird gelöscht, das heißt mit 0-Bytes bzw. mit der Paper-Farbe gefüllt.

ART,x
Für x sind hier nur 2 Werte zulässig: 0 und 1. Dies ist ein grundlegender Befehl für die Window-Verwaltung. ART,0 schaltet in den Normalzustand, das heißt, alle Befehle funktionieren so, wie bisher beschrieben. ART,1 dagegen bewirkt, daß um ein Fenster immer ein Rahmen gezogen wird, der die Grenzen des Fensters sichtbar macht. Dies ist besonders günstig, wenn mehrere Fenster sich gegenseitig überlappen. Die beim Open-Befehl angegebene Ausdehnung des Fensters bezeichnet aber nun nicht die tatsächliche Größe des Fensters, sondern nur die von dem Rahmen umgebene Fläche. Dabei ist noch zu beachten, daß diese Fläche gleichzeitig gelöscht wird und nicht, wie im Normalzustand, auf dem Bildschirm erhalten bleibt. Ebenso löscht der CLS-Befehl nicht das ganze Window, sondern nur die Innenfläche. Dagegen bleiben das CLOSE-, SHOW-, und CLEAR-Kommando unbeeinflußt. CLOSE und CLEAR beachten unabhängig davon, daß das ART-Kommando zwischenzeitlich benutzt sein könnte, immer nur den von dem Window überdeckten Bereich. Weitere Auswirkungen des Art-Kommandos werden aufgrund ihrer Komplexität in den einzelnen Kommando-Erklärungen beschrieben.

HEADLINE,w,s
Dieser Befehl ist zum Betiteln der bereits geöffneten Windows gedacht. Der String s wird zentriert und von Strichen flankiert in die oberste Zeile des Fensters geschrieben. Dabei ist zu beachten, daß die Schriftart durch das INVERS- und KURSIV-Kommando variiert werden kann. Weiterhin ist anzumerken, daß beim Darstellungsmodus 1 (ART,1) unter der obersten Zeile nun noch ein Strich gezogen wird, der diese "Überschrift" von der Restfläche des Fensters abtrennt. Außerdem wird diese neue Fläche als Schreibfläche wie beim RANGE-Kommando definiert.
Beispiel: **ART,1:OPEN,1,10,30,9,50: HEADLINE,1," MENUE "**

Befehle zum Print-Komplex

PRINT,s
Da die Befehlserweiterung eine neue Koordinatenmatrix definiert, ist es nur schwer bzw. sehr umständlich, durch Kombination von TAG- und MOVE-Kommandos ein Fenster zu beschriften, da diese ja in der Vertikalen beliebig definiert sein können. Daher wurde dieser Befehl geschaffen, der die neuen Grenzen eines Windows beachtet. Zu Beginn ist die aktuelle Print-Position (0,0). Jeder neue PRINT-Befehl beginnt an der aktuellen Print-Position. Die bekannten Vereinbarungen, wie Strichpunkt, Komma etc., hinter einem normalen Print-Kommando, gelten nicht mehr. Dagegen bewirkt ein CHR$(1) innerhalb des Strings, über die Tastatur mit CTRL-A einen Zeilenvorschub zu erreichen, das heißt, die aktuelle Position befindet sich danach in der nächsten Zeile in der ersten Spalte, je nach Definition mit RANGE. Sollen Zahlen dargestellt werden, so müssen diese erst mit STR$ oder DEC$ in Strings umgewandelt werden. Mehrere Strings werden nicht mit ";" sondern mit "+" verkettet, was beim originalen Print-Kommando auch möglich ist.

RANGE,w
Die aktuelle Print-Position kann nur Werte annehmen, die sich innerhalb eines definierten Feldes befinden. Dieses Feld wird von diesem Kommando definiert, das heißt, in diesem speziellen Fall kann nicht außerhalb des durch den OPEN-Befehl definierten Bereich geschrieben werden. Die Feldgröße wird also gemäß eines bereits definierten Fensters eingerichtet. Außerhalb exisiert nach dem RANGE-Befehl auch ein normales Bildschirmfenster, in das durch den normalen Print-Befehl geschrieben werden kann.

PRINT,w,s
Dieses Kommando ist gleichwertig mit der Befehlsfolge RANGE,w:PRINT,s.
Beispiel: **PRINT,1,"CPC664"**

RANGE,x1,y1,x2,y2
Gegenüber dem RANGE-Befehl mit nur einem Parameter besteht nur der Unterschied, daß die Fenstergrenzen nicht von einem bereits bestehenden Window bestimmt werden, sondern durch die angegebenen Parameter.

RANGE (ohne Parameter)
Die Schreibgrenzen werden auf den gesamten Bildschirm ausgedehnt, daher ist dieser Befehl gleichwertig mit
RANGE,0,0,80,200.
Beachte: RANGE wirkt sich auch auf den normalen Print-Befehl aus!

LOCATE,x,y
Positionierung der aktuellen Schreibposition auf die angegebene Stelle bezüglich des neuen, durch den RANGE-Befehl definierten Ursprung (obere linke Ecke des Fensters).

LOCATE,0,0
weist auf diesen Punkt.

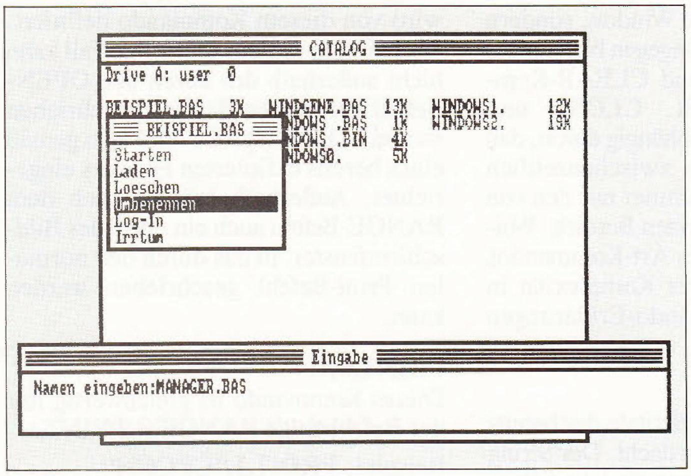
XTAB,x
Die aktuelle Printposition wird um x Stellen nach rechts verschoben.

YTAB,y
Die Printposition wird um y Pixel nach unten verschoben.

KURSIV (ohne Parameter)
Die Schriftart des neuen Print-Kommandos wird auf kursiv umgestellt. Dieser, wie auch die folgenden zwei Befehle, haben auch Auswirkungen auf das Schriftbild des MENÜ- und HEADLINE-Kommandos.

NORMAL (ohne Parameter)
Die Schrift wird auf den normalen Zeichensatz umgestellt. Keine Auswirkungen auf das INVERS-Kommando!

INVERS (ohne Parameter)
Die Schrift wird invertiert. Zweimaliges Aufrufen dieser Funktion hat wieder normales Schriftbild zur Folge.



Die Hardcopy zeigt nur eine Anwendungsmöglichkeit

Besondere Befehle:

MODE64,z
Wenn Sie einen Zusatzspeicher besitzen (und das ist beim CPC 6128 der Fall), können Sie die Lage des Stacks auch auf den normalen Speicherbereich umbiegen. Dies ist z.B. sinnvoll, wenn Sie die zweite Speicherbank anders nutzen möchten (RAM-Disc oder ähnliches). In dem Argument z geben Sie die unterste Speicherstelle des Stacks an. Für einen Besitzer des CPC 464 oder CPC 664 ist dieser Befehl natürlich enorm wichtig, da dieser Befehl ihm das Betreiben dieser Zusatzbefehle ohne Speichererweiterung möglich macht. Hinweis: Beim Einbinden der Erweiterung wird dieser Befehl automatisch durchgeführt, wenn das Programm feststellt, daß keine zweite RAM-Bank zur Verfügung steht. Für z wird der Wert &4000 voreingestellt, diesen können Sie aber nach Belieben mit diesem Befehl umändern. Beachten Sie, daß auf jeden Fall HIMEM heruntergesetzt werden muß. Also:
MEMORY z-1:MODE64,z.
In der "Einbindemeldung" nach dem **CALL &8A00** bekommen Sie auch Informationen, wie der Speicherbereich voreingestellt worden ist, bzw. ob Sie einen Zusatzspeicher besitzen oder nicht. Diese Meldung können Sie übrigens unterbinden, wenn Sie vor dem CALL in die Adresse &A67A den Wert 144 einPOKEn.

MODE128 (ohne Parameter)
Dies ist, wie man unschwer erraten kann, der gegenteilige Befehl zum vorherigen. Mit **MODE128** weist man den Interpreter an, die zweite RAM-Bank als Stack zu benutzen. Da diese ganz genutzt wird, braucht keine Speichergrenze angegeben zu werden.

GRID (ohne Parameter)
Dieser Befehl veranlaßt, daß der gesamte Bildschirm mit einem schachbrettartigen Pixelmuster gefüllt wird. Da die Struktur dieses Musters sehr fein ist, erscheint der Bildschirm nun in einer Mischfarbe aus Pen- und Paperfarbe. Vor einem solchen Hintergrund sehen die Windows in Darstellungsmodus 1 (ART,1) besonders gut aus.

Sehr mächtige Befehle

CURSOR,vx,vy
An der Koordinate (x,y) erscheint ein kleiner Pfeil, der sich durch Cursortasten und den Joystick bewegen läßt. Durch Drücken von Enter oder dem Feuerknopf wird die Funktion abgebrochen, der Pfeil verschwindet wieder und die neuen Koordinaten werden in x und y übergeben. Dieser Befehl läßt sich z.B. zum Anwählen von Icons benutzen. Nicht die "Klammeraffen" vor den Variablen und das vorherige Definieren derselben vergessen!
Beispiel: **d%=40:e%=100:CURSOR, d%, e%:PRINT d%;e%**

GETTEXT,s,x,y
Achtung: s darf hier nur eine String-Variable sein, diese muß auch einen Inhalt haben. An der Stelle (x,y) wird aus dem Bildschirm ein Text gelesen und in die Variable s übergeben. Der Text/das Wort wird nur in der Länge eingelesen, wie die String-Variable vorher definiert worden ist. Außerdem muß sich der zu lesende Text auf einer ganz normalen Print-Position befinden. Wenn sich in dem zu lesenden Text auf dem Bildschirm eine Leerstelle befindet, bricht die Funktion ab und übergibt den bis dahin gelesenen Text. Wenn sich zu Beginn an der zu lesenden Stelle nichts befindet, bzw. eine Leerstelle ist, wird links von dieser Stelle nach einem Zeichen gesucht. Wird eins gefunden, so wird angenommen, es sei ein Wort und dessen Anfang wird gesucht. Ab dieser Stelle wird dann wieder angefangen zu lesen. Beispiel: Wenn gewünscht wird, daß das eingelesene Wort die Länge von zehn Buchstaben hat, so definiert man:
a$=space$(10):GETTEXT, s,x,y

MENUE,vw,x,y,z,s,s1,s2,s3,....,s23
Dieser Befehl hat nicht nur eine einfache Funktion, sondern ist fast schon ein Unterprogramm. An der Koordinate (x,y) wird ein Fenster von der Breite z aufgebaut. Für diesen einen Befehl wird in **ART,1** umgeschaltet, nach Ausführung des Befehls ist wieder die vorherige Einstellung gültig. Eine Headline am oberen Rand des Fensters trägt die Bezeichnung s. Darunter werden die Strings s1, s2 usw. aufgeführt. Theoretisch lassen sich so bis zu 23 Menüpunkte auflisten, jedoch erhält man, je nach Stringgröße, die Fehlermeldung *'String expression too complex'.* Beachten Sie, daß Sie mindestens einen Menüpunkt angeben müssen, das heißt, der Befehl MENÜ benötigt mindestens sechs Argumente. Die Höhe des Fensters wird automatisch so eingestellt, daß alle Menüpunkte genau in das Fenster passen. Nun erscheint der oberste Menüpunkt invertiert. Dieser Balken über den Menüpunkten läßt sich mit Space, dem Joystick oder den Cursortasten bewegen. Haben sie einen Punkt angewählt, können Sie die Funktion wieder durch Druck auf Enter oder die Feuertaste verlassen. Die Nummer des Menüpunktes erscheint in w. Diese Variable muß natürlich wieder vom Typ Integer sein, vorher definiert werden, und einen "Klammeraffen" vorangestellt bekommen. Nach Ausführung des Befehls verschwindet das Fenster nicht (siehe nächster Befehl, Beispiel dort).

CLOSEMENÜ (ohne Parameter)
Dieser Befehl schließt das zuletzt eröffnete Menüfenster wieder, das heißt, der vorherige Bildschirminhalt wird wieder sichtbar. Da angenommen wird, daß Menüpunkte sowieso hierarchisch aufgebaut werden, wurde auf eine Numerierung derselben verzichtet. Daher lassen sich die Fenster auch nur in der umgekehrten Folge ihres Erscheinens schließen. Intern erhalten die Menüfenster die Nummern 200-255. Mit dem Befehl CLOSE (ohne Parameter) werden auch die Menüfenster geschlossen, diese noch vor den anderen, normal definierten Fenstern.
Beispiel: **r%=0:MENUE, r%,30,20, 10," TEST ","1. Wahl","2. Wahl": CLOSEMENUE:PRINT r%**

SAVE,x,s
Beim Öffnen eines Fensters wird der darunterliegende, bzw. verdeckte Bereich im Stack abgelegt. Wenn nun die Befehle *OPEN-SHOW-CLOSE-CLEAR* für grafische Anwendungen benutzt werden, z.B. um Icons zu handhaben, entsteht der Wunsch, diese Grafiken auch auf Diskette abspeichern zu können. Dieser Wunsch kann mit diesem Befehl erfüllt werden. x gibt die Nummer des Fensters

an, das natürlich schon existieren muß. s ist der Name, den das File erhält. Halten Sie sich beim Namen an die üblichen Bestimmungen, acht Buchstaben Name, Punkt, drei Buchstaben Extension, die aber auch weggelasssen werden kann. Wundern Sie sich aber nicht, wenn Sie für einen Ausschnitt von Bildschirmgröße nur 3 bis 4 Kilobyte auf der Diskette benötigen: es wurde ein Kompressor eingebaut, der die Grafikdaten komprimiert! Beispiel siehe LOAD.

LOAD,x,s
Mit LOAD können Sie die durch SAVE abgespeicherten Grafikdaten wieder in den Stack (aber noch nicht auf den Bildschirm) bringen. Dazu ist es nicht nötig, ein Fenster mit der Nummer s definiert zu haben. Existiert ein solches, wird es erst einmal wieder geschlossen. Daraufhin wird ein Fenster mit der Nummer x und den Größen festgelegt, wie es einst (vor dem Abspeichern) definiert worden ist. Wenn Sie diese Daten haben wollen, können Sie diese mit dem Befehl **GETLOC** erfahren. Soll nun die Grafik auf dem Bildschirm erscheinen, erreichen Sie dies wie gewohnt mit dem Befehl
CLOSE,x bzw SHOW,x.
Anwendungsbeispiel: Sie haben eine Grafik auf dem Bildschirm, die Sie weiterverarbeiten wollen. Nun definieren Sie ein Fenster so, daß die entsprechende Grafik überdeckt wird. Möglichst dabei in **ART,0** arbeiten. In diesem Fall haben Sie zwar keine Veränderung auf dem Bildschirm, aber eine Kopie des Motives im Stack. Dieses können Sie nun mit **SAVE** auf Diskette legen und das auch noch komprimiert. **LOAD** befördert diese Grafik wieder zurück in den Speicher. Mit Hilfe von **SHOW** kann das Motiv z.B. als Icon an verschiedene Stellen auf den Bildschirm gebracht werden.
Beispiel: **ART,0 OPEN,1,10,40,10,50** (an dieser Stelle des Bildschirms sollte irgendetwas stehen)
SAVE,1,"testbild"
CLEAR
LOAD,7,"testbild"
SHOW,7,40,100
SHOW,7,30,150

Installation der Befehlserweiterung

Tippen Sie zunächst den Data-Lader *"WINDGENE.BAS"* ab und starten ihn. Sollten Sie keine Fehler gemacht haben, so wird nach einiger Zeit das File *"WINDOWS.BIN"* auf Ihrer Diskette erzeugt. Dieses File muß nach &8A00 geladen werden und daraufhin werden mit **CALL &8A00** die neuen Befehle installiert. Die gleiche Aufgabe übernimmt das Programm *"WINDOWS.BAS"*. Die oberste Speicherstelle von Window-Manager liegt noch unter &A000, das heißt, andere Befehlserweiterungen, die meist über dieser Speicherstelle liegen, können eingebunden werden. Es lohnt sich, das Beispielprogramm *"Beispiel.Bas"* abzutippen, Sie werden staunen, was dieses kleine Programm auf den Bildschirm zaubert! Es handelt sich um eine menü- und mauszeigergesteuerte Benutzeroberfläche!

Anleitung zum Beispielprogramm

Das Programm besteht aus zwei Teilen. Nach Starten des Programmes befinden Sie sich im ersten Teil, der das Management von Diskettenfiles übernimmt. Bewegen Sie den Mauszeiger auf irgendeinen Filenamen und drücken Sie Enter, Copy oder Fire. Ein Pulldown-Menü wird heruntergeklappt, die einzelnen Menüpunkte verstehen sich von selbst. Sollten sie mit der Option LADEN ein Maschinensprachefile eingeladen haben, und bricht der Rechner nicht ab, so gelangen Sie in den Grafik-Manager (es wurde nämlich angenommen, das Binärfile sei eine Bildschirmgrafik gewesen). Eine andere Möglichkeit, in den Grafik-Manager zu gelangen, haben Sie, wenn Sie den Mauszeiger anklicken, wenn er nicht auf einen Filenamen zeigt. In diesem Fall gelangen Sie in den Grafik-Manager, wenn Sie die entsprechende Option anwählen. Der Grafik-Manager selbst bietet die Möglichkeit, Screens einzuladen und abzuspeichern, Ausschnitte auszuschneiden und an anderer Stelle zu kopieren oder diese auf Diskette (später vielleicht als Icons zu gebrauchen) abzuspeichern. (Das entsprechende Menü erhalten Sie, wenn Sie den Mauszeiger irgendwo anklicken.) Wenn Sie einen Ausschnitt definieren wollen, bewegen Sie den Mauszeiger auf eine Ecke des gewünschten Feldes und drücken Enter (oder Copy oder Fire). Daraufhin wählen Sie "definieren" an und bewegen den Mauszeiger auf die gegenüberliegende Ecke, wieder gefolgt von Enter. Kopieren funktioniert ähnlich: Ausschnitt definieren, Mauszeiger auf gewünschte Stelle, Enter, "kopieren" anwählen. "löschen" löscht nur den Bereich auf dem Bildschirm, nicht den gespeicherten Bereich. Bedenken Sie, daß dieses Programm nicht unbedingt von großen Nutzen für Ihre künstlerischen Fähigkeiten sein soll, sondern nur die Möglichkeiten der Befehlserweiterung demonstrieren soll.

(Michael Christen/cd)

für 464-664-6128

```
10 REM "WINDOWS.BAS" Basic-Lader / Install [2131]
ation
20 IF HIMEM>35327 THEN MEMORY &89FF:LOAD " [5084]
windows.bin":CALL &8A00
30 'Ihr Programm...                        [863]
```

```
1 REM "WINDGENE.BAS", Data-Lader fuer Wind [5823]
ow-Manager                 1987
                                 by Micha
el Christen
2 MEMORY &89FF:RESTORE 10:z=10:FOR n=&8A00 [4097]
 TO &9796 STEP 21:x=0
3 FOR m=0 TO 20:READ a$:a=VAL("&"+a$):POKE [4312]
 n+m,a:x=x+a:NEXT m
4 READ a$:IF x<>VAL("&"+a$) THEN PRINT"DAT [2947]
A ERROR in ";z:END
5 z=z+1:NEXT n:SAVE "windows.bin",b,&8A00, [3352]
&D97,&8A00
6 PRINT"OK. Binaerfile abgespeichert.":END [5146]
10 DATA 21,cf,8a,01,d3,8a,cd,d1,bc,00,00,0 [3056]
0,cd,af,8b,21,aa,96,22,a8,96,09fa
11 DATA 21,ff,3f,cd,55,95,f5,e5,3e,3a,cd,5 [4839]
2,95,21,ff,7f,cd,55,95,fe,3a,0baa
12 DATA 20,0c,11,01,40,cd,d0,8b,21,d0,96,2 [4193]
2,a8,96,e1,f1,cd,52,95,21,40,0974
13 DATA 96,11,96,96,06,10,7e,ee,44,12,23,1 [1950]
3,10,f8,3a,7a,a6,fe,90,28,53,084c
14 DATA 3e,90,32,7a,a6,cd,11,bc,f5,cd,a7,8 [3289]
a,3e,01,01,50,c8,11,00,00,cd,08e3
15 DATA 0a,8c,3e,02,cd,0e,bc,cd,fc,94,21,0 [4239]
2,0a,22,79,97,22,77,97,21,be,0838
16 DATA 4d,22,7b,97,3e,2f,32,65,92,11,50,9 [3264]
6,cd,2d,92,ed,5b,a8,96,cd,2d,091a
17 DATA 92,11,f3,96,cd,2d,92,cd,d7,92,cd,1 [4596]
8,bb,f1,cd,0e,bc,cd,4b,8d,21,0bdc
18 DATA 00,00,11,5d,97,01,03,00,cd,ce,95,2 [2989]
1,9f,97,22,83,97,af,32,82,97,07c6
19 DATA 21,00,00,22,77,97,22,79,97,21,50,c [4160]
8,22,7b,97,c3,c3,95,00,00,00,070b
20 DATA 00,23,8b,c3,af,8b,c3,cd,8b,c3,ab,8 [3823]
d,c3,ef,8b,c3,f7,8b,c3,58,8d,0ceb
21 DATA c3,e0,8d,c3,32,8d,c3,84,90,c3,eb,9 [3139]
1,c3,00,92,c3,13,92,c3,1e,92,0bf8
22 DATA c3,29,92,c3,9c,92,c3,a8,92,c3,d7,9 [3210]
2,c3,f7,92,c3,4a,93,c3,fc,94,0dd7
23 DATA c3,f3,93,c3,13,8e,c3,cd,8e,c3,1d,9 [4203]
5,c3,a0,8f,c3,61,94,4d,4f,44,0bca
24 DATA 45,31,32,b8,4d,4f,44,45,36,b4,43,4 [3719]
c,45,41,d2,41,52,d4,4f,50,45,07a1
25 DATA ce,43,4c,4f,53,c5,53,48,4f,d7,4e,4 [4061]
5,57,50,4f,d3,4d,45,4e,55,c5,08db
26 DATA 43,4c,4f,53,45,4d,45,4e,55,c5,4c,4 [4266]
f,43,41,54,c5,58,54,41,c2,59,07b0
27 DATA 54,41,c2,50,52,49,4e,d4,4b,55,52,5 [2710]
3,49,d6,4e,4f,52,4d,41,cc,49,085a
28 DATA 4e,56,45,52,d3,43,4c,d3,48,45,41,4 [3449]
4,4c,49,4e,c5,47,52,49,c4,52,0822
29 DATA 41,4e,47,c5,53,41,56,c5,4c,4f,41,c [2919]
4,47,45,54,4c,4f,c3,43,55,52,0812
30 DATA 53,4f,d2,47,45,54,54,45,58,d4,00,2 [2666]
1,84,95,22,53,95,21,96,95,22,07cb
31 DATA 56,95,21,00,00,22,61,97,21,ff,ff,2 [3744]
2,63,97,22,5d,97,c3,a7,8a,fe,0969
32 DATA 01,c0,21,6e,95,22,53,95,21,79,95,2 [4269]
2,56,95,ed,53,61,97,21,ff,89,090c
33 DATA af,ed,52,22,63,97,22,5d,97,c3,a7,8 [4377]
a,fe,01,c0,7b,32,81,97,c9,fe,0b5f
34 DATA 05,c0,dd,7e,08,fe,00,c8,43,dd,4e,0 [3648]
2,dd,56,04,dd,5e,06,32,6b,97,090a
35 DATA 3a,81,97,fe,00,28,07,04,04,04,04,0 [3907]
4,0c,0c,ed,43,6c,97,ed,53,6e,068c
36 DATA 97,3a,6b,97,c5,cd,60,8d,cd,ee,8c,c [4305]
1,cd,ec,95,54,5d,cd,e5,8c,3a,0cd1
37 DATA 6b,97,f5,d5,cd,ac,8c,cd,63,95,ed,5 [2780]
3,73,97,d1,cd,58,95,ed,53,75,0d20
38 DATA 97,23,23,f1,cd,52,95,23,ed,4b,6e,9 [3957]
7,50,59,cd,58,95,e5,cd,fa,95,0b86
39 DATA eb,e1,d5,23,23,ed,5b,6c,97,42,4b,c [2461]
d,58,95,23,23,d1,c5,06,00,d5,0a30
40 DATA cd,ce,95,d1,e5,62,6b,cd,26,bc,54,5 [3441]
d,e1,c1,10,ec,e5,2a,73,97,ed,0cb7
41 DATA 5b,75,97,af,ed,52,eb,e1,cd,58,95,2 [4298]
3,23,af,cd,52,95,cd,c3,95,3a,0be3
```

Listing WINDOWS

```
42 DATA 81,97,fe,00,c8,3a,6b,97,cd,fb,92,c [3375]
9,21,00,00,cd,e5,8c,d5,cd,63,0ba1
43 DATA 95,42,4b,d1,23,23,cd,55,95,2b,2b,f [4610]
e,00,20,08,7a,b8,d8,20,03,7b,0814
44 DATA b9,d8,09,3a,64,97,bc,20,df,3a,63,9 [3513]
7,bd,20,d9,f1,f1,f1,cd,c3,95,0c6c
45 DATA 11,33,97,c3,27,96,e5,21,07,00,19,5 [3865]
4,5d,e1,c9,21,00,00,cd,63,95,07c2
46 DATA 23,23,cd,55,95,2b,2b,e5,d5,f5,19,3 [3090]
a,64,97,bc,20,0c,3a,63,97,bd,0929
47 DATA 20,06,f1,d1,e1,c3,c3,95,f1,fe,00,2 [4586]
0,18,23,23,cd,55,95,2b,2b,fe,0a5c
48 DATA 00,20,0d,cd,63,95,00,e1,19,eb,e1,c [3938]
d,58,95,18,c3,d1,d1,18,bf,fe,0ac4
49 DATA 03,c0,dd,7e,04,cd,be,8d,fe,00,c8,2 [5104]
3,23,23,dd,56,00,dd,5e,02,cd,09a6
50 DATA 58,95,c9,3e,ff,f5,cd,60,8d,f1,3d,2 [2411]
0,f8,c3,a7,8a,fe,00,28,ef,fe,0cef
51 DATA 01,c0,7b,cd,be,8d,fe,00,ca,c3,95,2 [4275]
3,23,cd,55,95,af,cd,52,95,23,0af7
52 DATA 3a,65,97,fe,00,28,05,11,00,c0,18,0 [3960]
b,cd,63,95,42,4b,e5,cd,fa,95,08e8
53 DATA eb,e1,d5,23,23,cd,63,95,23,23,42,4 [3406]
b,d1,c5,06,00,d5,cd,dd,95,d1,0b00
54 DATA c1,e5,62,6b,cd,26,bc,54,5d,e1,10,e [1998]
c,c3,c3,95,fe,00,ca,a7,8a,7b,0c3f
55 DATA cd,be,8d,fe,00,c8,23,23,af,cd,52,9 [4366]
5,c9,21,00,00,32,ce,8d,cd,63,0a2e
56 DATA 95,e5,23,23,cd,55,95,fe,01,e1,c8,1 [2531]
9,3a,64,97,bc,20,ec,3a,63,97,0a69
57 DATA bd,20,e6,af,c9,fe,01,20,10,7b,f5,a [3345]
f,32,6d,8d,f1,cd,60,8d,3e,af,0b4d
58 DATA 32,6d,8d,c9,fe,03,c0,dd,7e,04,f5,d [3973]
d,4e,02,43,cd,fa,95,3e,01,32,0a47
59 DATA 65,97,22,7a,8d,f1,cd,e5,8d,af,32,6 [3971]
5,97,c9,fe,02,c0,af,32,70,97,0ba3
60 DATA d5,dd,7e,02,cd,be,8d,fe,00,d1,c8,e [4009]
5,1a,47,13,1a,6f,13,1a,67,11,0968
61 DATA d7,97,cd,8c,bc,e1,d0,f3,cd,e7,92,7 [4331]
9,cd,67,8e,78,cd,67,8e,7b,cd,0e2a
62 DATA 67,8e,7a,cd,67,8e,42,c5,d5,43,c5,c [4898]
d,55,95,e5,cd,67,8e,e1,23,c1,0c38
63 DATA 10,f3,d1,c1,10,ec,cd,b9,8e,cd,8f,b [4925]
c,fb,c9,32,72,97,3a,70,97,fe,0cfb
64 DATA 00,20,12,3a,72,97,cd,c1,8e,c2,95,b [3701]
c,32,71,97,3e,01,32,70,97,c9,091f
65 DATA 3a,71,97,47,3a,72,97,b8,28,11,78,c [4008]
d,95,bc,3a,70,97,cd,95,bc,af,0a61
66 DATA 32,70,97,c3,71,8e,3a,70,97,3c,32,7 [3761]
0,97,fe,ff,c0,3a,71,97,cd,95,0b12
67 DATA bc,3a,70,97,cd,95,bc,af,32,70,97,c [4162]
9,3a,70,97,fe,00,c8,18,e7,fe,0bd0
68 DATA 00,c8,fe,55,c8,fe,aa,c8,fe,ff,c9,f [3179]
e,02,c0,af,32,70,97,dd,7e,02,0d1e
69 DATA f5,1a,47,13,1a,6f,13,1a,67,11,d7,9 [3375]
7,cd,77,bc,d2,7a,8f,ca,7a,8f,09b3
70 DATA fe,16,c2,7a,8f,f1,f5,cd,60,8d,cd,7 [3145]
d,8f,5f,cd,7d,8f,57,cd,7d,8f,0cc0
71 DATA 4f,cd,7d,8f,47,ed,43,6c,97,ed,53,6 [3649]
e,97,c5,cd,ee,8c,c1,cd,ec,95,0d02
72 DATA 54,5d,cd,e5,8c,d5,cd,ac,8c,f3,cd,6 [4242]
3,95,ed,53,73,97,d1,cd,58,95,0d56
73 DATA ed,53,75,97,23,23,f1,cd,52,95,23,e [4924]
d,5b,6e,97,cd,58,95,23,23,ed,0a94
74 DATA 5b,6c,97,cd,58,95,23,23,42,c5,d5,4 [4747]
3,c5,e5,cd,7d,8f,e1,cd,52,95,0b95
75 DATA 23,c1,10,f3,d1,c1,10,ec,e5,cd,7a,b [3102]
c,2a,73,97,ed,5b,75,97,af,ed,0c81
76 DATA 52,eb,e1,cd,58,95,23,23,af,cd,52,9 [5423]
5,fb,c3,c3,95,f1,fb,c9,3a,70,0cf6
77 DATA 97,fe,00,ca,8d,8f,21,70,97,35,3a,7 [3782]
1,97,c9,cd,80,bc,cd,c1,8e,c0,0bc8
78 DATA 32,71,97,cd,80,bc,32,70,97,c3,85,8 [2961]
f,fe,02,c0,1a,47,dd,6e,02,dd,0a9e
79 DATA 66,03,4e,ed,43,85,97,cd,3a,90,cd,a [4490]
c,91,f5,cd,6f,90,f1,ed,4b,85,0c13
80 DATA 97,fe,f0,20,15,78,fe,04,38,09,3d,3 [2928]
d,3d,3d,32,86,97,18,df,3e,c0,08ad
81 DATA 32,86,97,18,d8,fe,f1,20,14,78,fe,b [4170]
d,30,09,3c,3c,3c,3c,32,86,97,090d
82 DATA 18,c6,af,32,86,97,18,c0,fe,f2,20,1 [3922]
2,79,fe,00,28,06,3d,32,85,97,0906
83 DATA 18,b1,3e,4f,32,85,97,18,aa,fe,f3,2 [3261]
0,11,79,fe,4f,28,06,3c,32,85,086f
84 DATA 97,18,9b,af,32,85,97,18,95,fe,e0,2 [4352]
0,91,3a,86,97,dd,6e,00,dd,66,0a68
85 DATA 01,77,23,af,77,3a,85,97,dd,6e,02,d [4612]
d,66,03,77,23,af,77,c9,ed,4b,096b
86 DATA 85,97,cd,fa,95,e5,11,97,97,06,08,7 [3151]
e,12,cd,26,bc,13,10,f8,e1,e5,0aca
```

Listing WINDOWS

```
87 DATA 11,87,97,06,08,1a,a6,77,cd,26,bc,1 [2401]
3,10,f7,e1,11,8f,97,06,08,1a,077d
88 DATA b6,77,cd,26,bc,13,10,f7,c9,ed,4b,8 [2777]
5,97,cd,fa,95,11,97,97,06,08,0abc
89 DATA 1a,77,cd,26,bc,13,10,f8,c9,fe,06,d [4465]
8,d6,05,32,68,97,47,dd,23,dd,0a30
90 DATA 23,10,fa,dd,e5,cb,27,cb,27,cb,27,c [2651]
6,0e,47,dd,4e,02,dd,56,04,dd,0a21
91 DATA 5e,06,c5,d5,3e,c9,f5,cd,be,8d,fe,0 [3439]
0,28,0c,f1,3c,fe,ff,20,f2,d1,0c51
92 DATA c1,f1,dd,e1,c9,21,82,97,34,f1,2a,8 [3693]
3,97,77,23,22,83,97,d1,c1,cd,0c11
93 DATA 0a,8c,dd,e1,dd,5e,00,dd,56,01,3a,6 [2566]
b,97,cd,50,93,af,32,67,97,3e,09cc
94 DATA 01,32,66,97,dd,2b,dd,2b,2a,79,97,2 [3529]
2,77,97,dd,22,69,97,06,00,04,07b3
95 DATA dd,5e,00,dd,56,01,c5,3a,78,97,f5,c [2983]
d,2d,92,3a,79,97,32,77,97,f1,0a79
96 DATA 06,08,80,32,78,97,dd,2b,dd,2b,c1,3 [4169]
a,68,97,b8,20,d9,3a,67,97,fe,09c0
97 DATA 00,28,03,cd,82,91,3a,66,97,32,67,9 [3783]
7,cd,82,91,cd,ac,91,fe,f0,28,0a72
98 DATA 19,fe,e0,28,26,fe,f1,20,f1,3a,68,9 [3779]
7,21,66,97,be,28,03,34,18,2b,08fc
99 DATA 36,01,c3,78,91,21,66,97,3e,01,be,2 [3021]
8,03,35,18,1b,3a,68,97,77,18,0679
100 DATA 15,3a,68,97,dd,2a,69,97,dd,6e,0a, [3449]
dd,66,0b,3a,66,97,77,23,af,77,08ef
101 DATA c9,3a,68,97,dd,2a,69,97,c3,1f,91, [4578]
3d,cb,27,cb,27,cb,27,ed,4b,79,0a40
102 DATA 97,80,47,c5,cd,fa,95,c1,3a,7b,97, [2785]
91,4f,06,08,c5,e5,41,7e,2f,77,0a89
103 DATA 23,10,fa,e1,cd,26,bc,c1,10,f0,c9, [4222]
cd,18,bb,fe,f0,c8,fe,0b,28,26,0bf4
104 DATA fe,f1,c8,fe,0a,28,22,fe,20,28,1e, [4506]
fe,f2,c8,fe,08,28,1a,fe,f3,c8,0c26
105 DATA fe,09,28,16,fe,e0,c8,fe,0d,28,12, [2769]
fe,58,28,0e,af,c9,3e,f0,c9,3e,0a69
106 DATA f1,c9,3e,f2,c9,3e,f3,c9,3e,e0,c9, [4146]
3a,82,97,fe,00,c8,3d,32,82,97,0c35
107 DATA 2a,83,97,2b,7e,22,83,97,c3,60,8d, [3408]
fe,02,c0,2a,79,97,dd,7e,02,85,09b5
108 DATA 32,77,97,7b,84,32,78,97,c9,fe,01, [4111]
c0,3a,77,97,83,32,77,97,c9,fe,0ada
109 DATA 01,c0,3a,78,97,83,32,78,97,c9,fe, [3795]
01,20,06,ed,4b,77,97,18,0b,fe,0923
110 DATA 02,c0,d5,dd,7e,02,cd,f8,93,d1,1a, [3648]
fe,00,c8,13,eb,5e,23,56,f5,d5,0b9c
111 DATA c5,1a,fe,01,20,06,c1,cd,88,92,18, [3860]
28,f5,cd,06,b9,f1,cd,a5,bb,eb,0b76
112 DATA cd,fa,95,e5,06,08,1a,00,77,cd,26, [4875]
bc,13,10,f7,e1,00,00,00,cd,09,0860
113 DATA b9,c1,0c,3a,7b,97,b9,cc,88,92,d1, [3841]
13,f1,3d,20,c4,ed,43,77,97,c9,0b6e
114 DATA 3a,79,97,4f,3e,08,80,47,3a,7c,97, [3757]
d6,08,b8,d0,3a,7a,97,47,c9,3e,08f2
115 DATA cd,32,6e,92,21,b3,92,22,6f,92,c9, [3048]
21,6e,92,06,03,af,77,23,10,fc,08d0
116 DATA c9,e5,06,02,cb,2e,cd,26,bc,10,f9, [3511]
06,03,cd,26,bc,10,fb,06,03,2b,085e
117 DATA cb,3e,23,cb,16,2b,cb,16,23,cd,26, [4296]
bc,10,f1,e1,c9,3a,65,92,fe,00,09c5
118 DATA 28,03,af,18,02,3e,2f,32,65,92,c9, [3840]
23,23,23,cd,63,95,d5,23,23,cd,0769
119 DATA 63,95,c1,23,23,c9,fe,01,c0,7b,cd, [3027]
be,8d,fe,00,c8,cd,e7,92,ed,53,0c66
120 DATA 6c,97,ed,43,77,97,cd,fa,95,42,c5, [3444]
43,e5,af,77,23,10,fc,e1,cd,26,0bf5
121 DATA bc,c1,10,f1,cd,c3,95,3a,81,97,fe, [2433]
00,c8,ed,4b,77,97,cd,fa,95,ed,0d4a
122 DATA 5b,6c,97,3e,ff,cd,ce,93,cd,ce,93, [4293]
01,0f,f0,cd,de,93,3e,ff,cd,ce,0d0d
123 DATA 93,cd,ce,93,c9,fe,02,c0,dd,7e,02, [3147]
32,6b,97,d5,cd,be,8d,cd,e7,92,0d0e
124 DATA 3a,81,97,fe,00,28,07,0c,1d,1d,04, [4115]
04,04,04,c5,05,cd,fa,95,06,04,0605
125 DATA cd,26,bc,3e,ff,cd,ce,93,10,f6,3a, [3579]
81,97,fe,00,28,13,cd,26,bc,cd,0b27
126 DATA 26,bc,2b,1c,1c,3e,ff,cd,ce,93,cd, [5143]
ce,93,1d,1d,01,4f,c7,ed,43,7b,09da
127 DATA 97,c1,7b,e1,c5,e5,56,92,cb,3f,81, [3487]
4f,ed,43,77,97,d1,cd,2d,92,c1,0c7c
128 DATA 3a,81,97,fe,00,20,04,3e,08,18,02, [3732]
3e,0c,80,47,c5,3a,6b,97,cd,f8,07ab
129 DATA 93,c1,ed,43,77,97,78,32,7a,97,c9, [2191]
f5,c5,e5,43,77,23,10,fc,e1,cd,0c4c
130 DATA 26,bc,c1,f1,15,c9,00,3e,02,ba,c8, [4022]
15,d5,e5,70,16,00,19,2b,71,e1,091f
```

Listing WINDOWS

```
131 DATA cd,26,bc,d1,18,ec,fe,01,20,1e,7b, [2785]
cd,be,8d,fe,00,c8,cd,e7,92,3a,0b9a
132 DATA 81,97,fe,00,28,1b,04,04,04,0c,15, [3299]
15,15,15,15,1d,1d,18,0e,fe,04,043c
133 DATA 20,3a,53,dd,5e,02,dd,46,04,dd,4e, [4509]
06,ed,43,77,97,ed,43,79,97,79,0939
134 DATA 83,32,7b,97,f5,78,82,32,7c,97,f5, [2862]
cd,f1,94,63,6a,25,f1,47,f1,4f,0bac
135 DATA cd,f1,94,7b,5a,57,15,15,1d,1d,cd, [3013]
66,bb,ed,4b,77,97,c9,21,00,00,0900
136 DATA 11,18,4f,cd,66,bb,ed,4b,77,97,c9, [3317]
fe,03,c0,cd,78,bb,e5,cd,69,bb,0c0c
137 DATA e5,d5,21,00,00,11,18,4f,cd,66,bb, [4869]
dd,5e,00,dd,56,01,dd,6e,02,dd,08da
138 DATA 66,03,4e,1a,47,e5,cd,f1,94,7a,cd, [3910]
72,bb,7b,fe,01,28,26,cd,6f,bb,0a82
139 DATA cd,60,bb,1d,fe,21,38,f0,fe,7f,30, [2736]
ec,1c,7b,fe,01,28,11,cd,6f,bb,0aab
140 DATA cd,60,bb,1d,fe,21,38,04,fe,7f,38, [4165]
ec,1c,1c,e1,7b,3d,77,4b,dd,5e,09cf
141 DATA 04,dd,56,05,1a,47,af,b8,28,1e,13, [4314]
1a,6f,13,1a,67,c5,e5,3e,20,77,06f9
142 DATA 23,10,fc,e1,c1,79,e5,cd,6f,bb,cd, [3629]
60,bb,e1,77,23,0c,10,f2,d1,e1,0c49
143 DATA cd,66,bb,e1,cd,75,bb,c9,59,1c,50, [3385]
cb,3a,cb,3a,cb,3a,14,c9,21,00,0a67
144 DATA c0,06,04,c5,3e,aa,cd,10,95,3e,55, [3548]
cd,10,95,c1,10,f2,c9,06,10,c5,0955
145 DATA 06,80,77,23,10,fc,c1,10,f6,c9,fe, [3758]
05,c0,dd,7e,08,cd,be,8d,fe,00,0af8
146 DATA c8,cd,e7,92,af,dd,66,07,dd,6e,06, [4896]
71,23,77,dd,66,05,dd,6e,04,70,0a6a
147 DATA 23,77,dd,66,03,dd,6e,02,73,23,77, [2997]
dd,66,01,dd,6e,00,72,23,77,c9,089e
148 DATA c3,84,95,c3,96,95,7b,cd,52,95,23, [3069]
7a,cd,52,95,2b,c9,cd,55,95,5f,0b54
149 DATA 23,cd,55,95,57,2b,c9,e5,d5,ed,5b, [3357]
61,97,19,d1,77,e1,c9,e5,d5,ed,0cd1
150 DATA 5b,61,97,19,d1,7e,e1,c9,f5,7c,07, [4586]
07,e6,03,cd,a6,95,f1,e5,cb,f4,0c6a
151 DATA cb,bc,77,e1,c9,7c,07,07,e6,03,cd, [3524]
a6,95,e5,cb,f4,cb,bc,7e,e1,c9,0d76
152 DATA e5,21,60,97,be,e1,c8,32,60,97,c5, [3261]
4f,e6,03,47,79,e6,1c,87,b0,f6,0b79
153 DATA c4,01,00,7f,ed,79,c1,c9,01,00,7f, [4221]
3e,c0,ed,79,32,60,97,c9,1a,cd,09f1
154 DATA 52,95,13,23,0b,af,b9,20,f5,b8,20, [4136]
f2,c9,cd,55,95,12,23,13,0b,af,08f1
155 DATA b9,20,f5,b8,20,f2,c9,21,00,00,c5, [2694]
d5,16,00,59,19,10,fd,d1,c1,c9,0a0c
156 DATA 2a,7d,97,ed,43,7d,97,7c,b8,20,0b, [3606]
2c,7d,b9,20,06,2a,7f,97,23,18,07e4
157 DATA 13,cd,0b,bc,84,67,c5,06,00,09,c1, [3155]
af,b8,28,05,cd,26,bc,10,fb,22,0897
158 DATA 7f,97,c9,3a,65,92,f5,3e,2f,32,65, [3845]
92,d5,11,26,97,cd,2d,92,d1,cd,0a68
159 DATA 2d,92,f1,32,65,92,c9,09,2d,27,2c, [3429]
25,21,28,64,07,2c,36,2d,37,30,05fa
160 DATA 21,2a,55,53,96,20,57,20,49,20,4e, [4601]
20,44,20,4f,20,57,20,20,2d,20,04ae
161 DATA 20,4d,20,41,20,4e,20,41,20,47,20, [3523]
45,20,52,20,20,20,56,20,31,2e,0410
162 DATA 30,20,01,01,43,6f,70,79,72,69,67, [3622]
68,74,20,20,28,43,29,20,31,39,0569
163 DATA 38,37,20,20,20,62,79,20,20,4d,69, [3464]
63,68,61,65,6c,20,43,68,72,69,0643
164 DATA 73,74,65,6e,01,01,aa,96,23,ad,96, [3266]
4d,65,6d,6f,72,79,20,45,78,70,0828
165 DATA 61,6e,73,69,6f,6e,3a,20,36,35,35, [3930]
33,35,20,42,79,74,65,73,20,66,0697
166 DATA 72,65,65,20,21,d3,96,53,74,61,6e, [3821]
64,61,72,64,20,4d,65,6d,6f,72,0837
167 DATA 79,3a,20,48,49,4d,45,4d,20,61,74, [3733]
20,26,34,30,30,30,20,30,f6,96,061e
168 DATA 47,72,61,70,68,69,63,20,4d,65,6d, [3581]
6f,72,79,20,66,72,65,65,2e,01,0748
169 DATA 01,01,70,72,65,73,73,20,61,6e,79, [3158]
20,6b,65,79,20,74,6f,20,63,6f,06f5
170 DATA 6e,74,69,6e,75,65,0a,29,97,01,2a, [3069]
45,52,52,4f,52,2a,3a,20,27,36,05f3
171 DATA 97,53,70,65,69,63,68,65,72,75,65, [3899]
62,65,72,6c,61,75,66,2c,20,4f,0820
172 DATA 50,45,4e,20,6e,69,63,68,74,20,6d, [4191]
6f,65,67,6c,69,63,68,2e,ff,ff,08ad
173 DATA 00,c0,00,80,ff,ff,00,01,00,01,00, [3369]
00,00,00,00,00,00,00,00,00,00,0340
```

Listing WINDOWS

```
174 DATA 00,00,00,00,00,00,00,50,c8,00,00, [2774]
00,c0,00,00,9f,97,00,00,0f,07,0324
175 DATA 03,01,00,81,c0,e0,00,60,70,78,7c, [3629]
18,0c,06,00,00,00,00,00,00,00,0413
```

```
10 REM "BEISPIEL.BAS" Beispielprogramm zum [6029]
 Window-Manager von Michael Christen
20 MODE 2:INK 0,13:BORDER 13:INK 1,0:KEY D [7054]
EF 22,0,ASC("\"),ASC("`"),150:KEY 150,CHR$
(13)+"run"+CHR$(34)+"beispiel"+CHR$(13)
30 IF HIMEM>35327 THEN MEMORY &89FF:LOAD " [6159]
windows.bin":POKE &A67A,144:CALL &8A00
40 ¦CLEAR:¦ART,1                           [801]
50 ¦OPEN,20,10,0,58,195                    [1020]
60 x%=12:y%=20:¦HEADLINE,20," CATALOG ":¦R [4334]
ANGE,20:CAT
70 ¦RANGE:a$=SPACE$(12)                    [1134]
80 ¦CURSOR,@x%,@y%:¦GETTEXT,@a$,@x%,@y%    [1815]
90 WHILE MID$(a$,9,1)<>"." AND x%>1:x%=x%- [3971]
1:¦GETTEXT,@a$,@x%,@y%:WEND
100 WHILE MID$(a$,9,1)<>"." AND x%<80-12 A [5956]
ND x%>0:x%=x%+1:¦GETTEXT,@a$,@x%,@y%:WEND
110 yy%=y%+7:IF y%>100 THEN yy%=y%-6*8-26  [2645]
120 IF x%=0 THEN r%=1:¦MENUE,@r%,x%,yy%,20 [14522]
," What's up ? ","Log-In","Grafik-Editor",
"Basic","CP/M","Irrtum":IF r%=1 THEN ¦CLOS
EMENUE:GOTO 50 ELSE IF r%=2 THEN 240 ELSE
IF r%=3 THEN GOSUB 230:END ELSE IF r%=4 TH
EN GOSUB 230:¦CPM ELSE ¦CLOSEMENUE:GOTO 80
130 r%=1:¦MENUE,@r%,x%,yy%,20," "+a$+" "," [5898]
Starten","Laden","Loeschen","Umbenennen","
Log-In","Irrtum"
140 IF r%=1 THEN GOSUB 230:RUN a$          [1855]
150 IF r%=2 THEN GOSUB 230:LOAD a$:GOTO 24 [2313]
0
160 IF r%=3 THEN ¦ERA,a$:¦CLOSEMENUE:¦CLS, [4288]
20:GOTO 60
170 IF r%=4 THEN GOSUB 200:¦REN,n$,a$:¦CLO [3135]
SE,21:¦CLOSEMENUE:GOTO 50
180 IF r%=5 THEN ¦CLOSEMENUE:¦CLS,20:GOTO  [2422]
60
190 IF r%=6 THEN ¦CLOSEMENUE:GOTO 80       [2410]
200 ¦OPEN,21,0,150,78,40:¦HEADLINE,21," Ei [3706]
ngabe "
210 PRINT:INPUT " Namen eingeben:",n$:IF n [3654]
$="" THEN ¦CLOSE,21:GOTO 200
215 ¦CLOSE,21:RETURN                       [1548]
230 ¦CLOSE:¦ART,0:¦OPEN,7,0,0,20,1:¦PRINT, [5134]
7,"Neustart: CTRL \":¦RANGE:PRINT:RETURN
240 'Graphik-Manager                       [666]
250 ¦CLOSE:¦ART,0:¦OPEN,1,25,75,29,24:¦CLS [13448]
,1:¦PRINT,1,"  Sie Befinden sich nun im"+C
HR$(1)+" G R A F I K - M A N A G E R"+CHR$
(1)+"    Bitte Taste druecken":WHILE INKEY
$="":WEND:¦CLOSE,1
260 ¦CURSOR,@x%,@y%:xx%=x%:yy%=y%          [2021]
270 ¦ART,1:¦MENUE,@r%,5,50,18," Funktionen [14771]
 ","definieren","loeschen","kopieren","tau
schen","abspeichern","einladen","Screen ab
speichern","Screen einladen","Hauptmenue":
¦ART,0
290 IF r%=1 THEN ¦ART,1:¦OPEN,13,25,100,30 [14142]
,9:¦PRINT,13,"Zweiten Eckpunkt bestimmen":
FOR n=1 TO 800:NEXT n:¦CLOSE,13:¦CLOSEMENU
E:¦CURSOR,@xx%,@yy%:¦ART,0:¦OPEN,77,MIN(x%
,xx%),MIN(y%,yy%),ABS(x%-xx%+1),ABS(y%-yy%
+1):¦CLS,77:¦SHOW,77:GOTO 260
300 IF r%=2 THEN ¦CLOSEMENUE:¦CLS,77:GOTO  [2818]
260
310 IF r%=3 THEN ¦CLOSEMENUE:¦SHOW,77,x%,y [3975]
%:GOTO 260
320 IF r%=4 THEN ¦CLOSEMENUE:xxx%=xx%:yyy% [10268]
=yy%:¦GETLOC,77,@xx%,@yy%,@xxx%,@yyy%:¦CLO
SE,77:¦OPEN,88,xx%,yy%,xxx%,yyy%:¦OPEN,77,
x%,y%,xxx%,yyy%:¦NEWPOS,88,x%,y%:¦CLOSE,88
:GOTO 260
330 IF r%=5 THEN ¦ART,1:GOSUB 200:¦ART,0:¦ [2113]
SAVE,77,n$:¦CLOSEMENUE:GOTO 260
340 IF r%=6 THEN ¦ART,1:GOSUB 200:¦ART,0:¦ [4362]
LOAD,77,n$:¦CLOSEMENUE:GOTO 260
350 IF r%=7 THEN ¦ART,1:GOSUB 200:¦ART,0:¦ [4565]
CLOSEMENUE:SAVE n$,b,&C000,&4000:GOTO 260
360 IF r%=8 THEN ¦ART,1:GOSUB 200:¦ART,0:¦ [5208]
CLOSEMENUE:LOAD n$,&C000:GOTO 260
400 ¦CLOSEMENUE:GOTO 40                    [1175]
```

Listing WINDOWS



Durch den Datendschungel

SOLVE – entwirren Sie den Maschinencode

Nehmen wir an, Sie haben ungefähr folgendes Problem: Die Geheimnisse der Assemblersprache haben Sie erfolgreich ergründet. Pop und Push, Jump und Inc sind Ihnen keine böhmischen Dörfer mehr. Auch einen Disassembler haben Sie sich besorgt. Und Sie brennen darauf, endlich das eingebaute BASIC Ihres Rechners anzuschauen. So gehen Sie auf die Reise: Disassembler, &C000, oberes ROM, fertig ausgedruckt. Nun, was "macht" das Programm? Sie stoßen auf den ersten CALL-Befehl. Also bei der Zieladresse nachschauen, was dort passiert. Hoppla wieder ein CALL! Jetzt wird's schwierig! Nachdem Sie dem fünften CALL gefolgt sind und herausgefunden haben, wozu die Routine dort dient, kommen wir jetzt endlich zum Thema.

Am Ende der Routine steht einfach RET, wenn Sie jetzt nur noch wüßten, wohin! Und wenn Sie es noch wissen, kommt kurz darauf in der aufrufenden Routine wieder ein RET, und Sie wissen sich vor lauter RETs nicht mehr zu retten! Man sollte ein Programm haben...

Ein Licht im Dunkeln!

..., das schön auf Papier ausdruckt, welche Zieladressen beispielsweise in ihrem BASIC-Interpreter von woher angesprungen werden. Damit sind Sie bei SOLVE gerade richtig! SOLVE untersucht MC-Programme, die im RAM, im ROM, in einem Expansion-ROM oder auf Diskette stehen können. Es filtert die Sprünge mit den jeweils zugehörigen Zieladressen heraus und gibt sie anschließend nach Zieladressen sortiert auf dem Drucker aus. Somit können Sie auf einen Blick feststellen, von woher eine bestimmte Routine angesprungen wird!

Wie SOLVE vorgeht!

Beim Aufruf von SOLVE werden Sie nach dem zu untersuchenden Bereich gefragt (RAM, ROM, Expansion-ROM oder Disk-Datei). Im Falle des Expansion-ROM geben Sie noch die gewünschte ROM-Nummer ein, im Falle einer Disk-Datei (nach Anzeige des CATalogs) den Namen der Datei. Nun werden die zu registrierenden Sprünge näher bestimmt. Sie haben die Möglichkeit, nur Sprünge in einen bestimmten Speicherbereich ausgeben zu lassen (z.B. nur die Aufrufe der Systemvektoren &B900-&BDF4, oder nur die Sprünge innerhalb des untersuchten Programms selber...), auch können Sie wählen, ob die kurzen Relativsprünge (+/−128 Bytes) überhaupt berücksichtigt werden sollen oder nicht. Bei Bearbeitung einer Datei von Diskette geht es nach diesen Abfragen bereits los: die Datei wird ins RAM ab &4000 geladen und untersucht. In den anderen Fällen können Sie noch die Start- und Endadresse des zu untersuchenden Bereichs bestimmen (z.B. das BASIC-ROM von &C000&D000), woraufhin der gewählte Speicherbereich ins RAM nach &4000 &7FFF kopiert wird. In jedem Fall steht also der Maschinencode während der Untersuchung im RAM ab &4000. Er darf &7FFF, also eine Länge von 16 kByte, nicht überschreiten.

Während der Untersuchung zeigt SOLVE die aktuelle Position im MC-Code in Schritten von &100 an (also z.B. &C000, &C100...), außerdem zum Schluß die ermittelte Anzahl der Sprungziele. Dann wird die Liste der Sprungziele sortiert und schließlich auf den Drucker ausgegeben.

Wie Sie vorgehen!

Zunächst tippen Sie den Datalader (Listing 2) ab und speichern ihn zur Sicherheit mit SAVE"SOLVE.LIN". Wenn Sie ihn mit RUN starten und in den eingetippten Zeilen keine Fehler mehr gefunden werden, erzeugt der Lader die fertige MC-Datei SOLVE.OBJ. Dann tippen Sie Listing 1 ab (was wohl keine Mühe bereiten dürfte) und speichern es mit



Solve – das Hilfsprogramm, um den Speicher Ihres CPC zu durchforsten

SAVE"SOLVE" ab. Und nun kann's losgehen: Untersuchen Sie beliebige Maschinenprogramme einfach durch den Aufruf RUN"SOLVE"!

Wie der Programmierer von SOLVE vorging!

Im Unterschied zu einem im März '88 in der PC AMSTRAD International veröffentlichten Beitrag untersucht SOLVE nicht alle Bytes des Maschinencodes nacheinander darauf, ob sie einen Sprungbefehl darstellen. Bei dieser Vorgehensweise könnte zum Beispiel folgender Fehler auftreten:

Angenommen, das untersuchte Programm enthält die Assembler-Befehlsfolge

LD B,&CD
LD C,&00

also die Bytes &06,&CD,&0E,&00, dann würde &CD als Call-Befehl an die Adresse &000E interpretiert, obwohl das Byte &CD mit einem Call überhaupt nichts zu tun hat! SOLVE umgeht diesen Fehler dadurch, daß in unserem Beispiel beim Lesen des Codes &06 der Lesezeiger sofort auf das übernächste Byte &0E gesetzt wird, denn auf &06 folgt immer ein Datenbyte.

Eine Ausnahme, bei der solche Fehler kaum zu vermeiden sind, bilden längere Datenfelder wie Texte oder Listen mit Sprungadressen. In diesem Fall sind die von SOLVE ausgegebenen "Pseudo"-Sprungziele anhand eines Disassembler-Listings auszuscheiden (aus diesem Listing ist oft erkennbar, daß es sich bei bestimmten Codes um Daten handeln muß).

Trifft SOLVE bei der Untersuchung auf einen Sprung, der in den gewählten Zielbereich geht, legt es die Zieladresse (2 Byte) in einer Liste ("Hauptliste") im Block 4 der zweiten RAM-Bank ab. Unmittelbar danach legt es einen Zeiger (2 Byte) auf eine zweite ("Aufruf") Liste, die von &44 im RAM bis &7FFF in Block 5 läuft. In ihr werden alle Sprünge gespeichert, die an diese Zieladresse gehen. Bei der späteren Ausgabe auf den Drucker wird jeweils eine Zieladresse aus der Hauptliste ausgedruckt, und dann, dem dahinter plazierten Zeiger in die Aufrufliste folgend, die zugehörigen Aufrufe.

Zuvor ist nur noch die Hauptliste zu sortieren. Dazu verwendet SOLVE das überaus schnelle Quicksort. Um einen Überlauf des Stacks zu vermeiden, wurde es iterativ realisiert, mit sofortiger Weiterbearbeitung des kleineren Bereichs nach einer Partition. Die Anzahl der sortierten Zieladressen wird hierbei ebenfalls angezeigt.

Noch etwas vergessen?

Bei den Abfragen bietet SOLVE jeweils eine voreingestellte Eingabe, die durch bloßes Drücken von RETURN gewählt wird. SOLVE kann zu jedem Zeitpunkt mit der ESC-Taste abgebrochen werden. Es läuft auf dem CPC 6128 unter Verwendung der zweiten Bank (64 kByte), außerdem auf dem CPC 464 und 664 mit dk'tronics Speichererweiterung (die beiden Rechner verfügen nicht über den Sprungvektor &BD5B zur Bankumschaltung, deshalb verwendet SOLVE einen direkten OUT-Befehl). Vortex muß diesmal leider passen!

(Helmut Steeb/jb)

für 6128

464/664 mit dk'tronics-Speichererweiterung

```
10 MEMORY &7FFF                              [150]
20 LOAD"solve.obj":CALL &8000:CALL 0         [2729]
```

```
10 ' SOLVE.LIN -- erzeugt SOLVE.OBJ          [1585]
12 ' Helmut Steeb 2.12.1988                  [1357]
15 ' Hardware-Voraussetzung SOLVE.OBJ:       [808]
16 ' - 6128 (verwendet 2. Bank)              [1603]
17 ' - 464/664 mit dk'tronics Speichererwe [3045]
iterung
18 ' - no Vortex                             [205]
20 DEFINT a-k:DEFSTR p-z:MODE 2:MEMORY &7F [2650]
FF
30 PRINT:PRINT"DATAs lesen..."               [1843]
40 FOR o=&8000 TO &8D42                      [1692]
50  READ s:b=VAL("&"+s)                      [699]
60  m=m+b:POKE o,b                           [113]
70 NEXT                                      [350]
80 IF m<>373143 THEN PRINT"Fehler in Datas [2139]
":END
90 PRINT"Ok! SOLVE.OBJ speichern..."         [2406]
100 SAVE "solve.OBJ",b,&8000,&D43            [1857]
110 END                                      [110]
120 DATA CD,24,80,CD,DF,80,D4,B0, 83,CD,F6 [1960]
,83,CD,ED,85,CD
130 DATA F6,86,2A,71,89,F9,21,40, 00,11,41 [2691]
,00,36,00,01,33
140 DATA 01,ED,B0,C9,3E,C9,32,00, 80,21,02 [1413]
,00,39,22,71,89
150 DATA 3E,02,CD,0E,BC,3E,20,CD, 5A,BB,3E [3169]
,96,CD,5A,BB,3E
160 DATA 9A,06,4C,CD,4F,88,3E,9C, CD,5A,BB [2526]
,21,02,02,CD,75
170 DATA BB,3E,95,CD,5A,BB,21,9C, 89,CD,D9 [2323]
,88,21,02,4F,CD
180 DATA 75,BB,3E,95,CD,5A,BB,21, 03,02,CD [3145]
,75,BB,3E,93,CD
190 DATA 5A,BB,3E,9A,06,4C,CD,4F, 88,3E,99 [2829]
,CD,5A,BB,21,EF
200 DATA 8A,CD,D9,88,CD,BA,BB,21, 8C,01,11 [3486]
,70,02,01,2A,08
210 DATA CD,9C,80,21,82,01,11,60, 02,01,16 [3007]
,10,D5,C5,58,16
220 DATA 00,CD,C0,BB,E1,D1,26,00, 22,92,89 [2259]
,ED,53,94,89,21
```

Listing SOLVE

```
230 DATA 00,00,CD,F9,BB,11,00,00, 2A,92,89 [2432]
,CD,D7,80,CD,F9
240 DATA BB,2A,94,89,CD,D7,80,EB, 21,00,00 [1319]
,CD,F9,BB,11,00
250 DATA 00,2A,92,89,C3,F9,BB,7C, 2F,67,7D [2419]
,2F,6F,23,C9,CD
260 DATA 97,81,F5,CD,72,81,CD,3D, 81,CD,3D [2208]
,81,F1,D8,CD,78
270 DATA BB,22,99,89,2A,99,89,CD, 75,BB,2A [3348]
,75,89,E5,2A,77
280 DATA 89,E5,CD,3D,81,E5,7C,D6, 40,57,5D [3884]
,ED,53,90,89,11
290 DATA C4,8A,CD,03,89,CD,3D,81, E5,11,EA [2463]
,8A,CD,03,89,E1
300 DATA D1,CD,24,83,CB,F4,22,73, 89,E1,22 [3212]
,77,89,E1,22,75
310 DATA 89,D0,CD,28,88,21,08,8A, CD,79,88 [2724]
,18,B7,2A,77,89
320 DATA CD,D9,88,CD,4B,88,E5,3E, 04,CD,39 [1943]
,83,CD,92,88,EB
330 DATA E1,38,EA,CD,2D,83,38,E5, 22,77,89 [2860]
,2A,75,89,28,0D
340 DATA 73,23,72,23,22,75,89,D5, CD,28,88 [2257]
,E1,C9,5E,23,56
350 DATA 18,F1,21,FA,8C,CD,D9,88, CD,4B,88 [4062]
,3E,01,CD,39,83
360 DATA CD,2D,83,38,ED,3E,30,28, 06,23,7E [2796]
,FE,32,30,E3,E6
370 DATA 01,32,9B,89,C3,28,88,21, 08,8B,CD [2431]
,D9,88,21,B7,8B
380 DATA CD,D9,88,CD,4B,88,3E,01, CD,39,83 [2626]
,CD,2D,83,38,ED
390 DATA 3E,34,28,02,23,7E,FE,34, CA,59,82 [2617]
,30,E0,FE,30,38
400 DATA DC,F5,CD,28,88,CD,28,88, F1,28,14 [1958]
,FE,32,28,7C,30
410 DATA 49,3E,FD,01,51,8B,DD,2E, 12,26,C0 [2951]
,16,FF,18,0C,26
420 DATA 00,16,3F,3E,FE,01,2F,8B, DD,2E,1B [2447]
,32,BC,83,2E,00
430 DATA 1E,FF,22,6D,89,D5,E5,11, A5,8C,60 [2353]
,69,DD,4D,06,00
440 DATA ED,B0,FE,FC,DC,3D,82,97, 12,E1,11 [2626]
,C4,8A,CD,03,89
450 DATA E1,22,6F,89,11,EA,8A,C3, 03,89,21 [2019]
,E5,8B,CD,D9,88
460 DATA CD,4B,88,3E,02,CD,39,83, CD,2D,83 [3030]
,38,ED,3E,07,28
470 DATA A8,CD,92,88,7D,01,95,8B, DD,2E,0E [3064]
,18,9C,F5,3E,26
```

Listing SOLVE

```
480 DATA 12,13,F1,CD,10,89,3E,20, 12,13,C9 [2701]
,3E,FF,26,A0,16
490 DATA A5,01,73,8B,DD,2E,0C,18, 92,CD,28 [2212]
,88,CD,E1,82,78
500 DATA B7,CA,12,80,CD,AA,82,30, F3,E5,D5 [3036]
,CB,F4,22,73,89
510 DATA 7A,D6,40,57,ED,53,90,89, 21,7B,89 [2999]
,4E,06,00,23,11
520 DATA A5,8C,ED,B0,3E,20,12,13, 97,12,21 [2149]
,0B,8C,CD,D9,88
530 DATA E1,22,6D,89,CD,F9,88,21, 52,8A,CD [2596]
,D9,88,E1,19,22
540 DATA 6F,89,CD,F9,88,CD,28,88, 37,C9,CD [2129]
,57,BB,CD,77,BC
550 DATA F5,CD,54,BB,F1,30,1D,D5, 11,18,00 [1997]
,19,D1,CD,65,88
560 DATA 2B,3E,40,BC,38,13,D5,E5, 21,00,40 [2348]
,CD,83,BC,CD,7A
570 DATA BC,E1,D1,D8,21,32,8A,18, 03,21,43 [3031]
,8A,CD,79,88,B7
580 DATA C9,CD,78,BB,E5,CD,6D,88, 21,00,C0 [3306]
,11,00,40,42,4B
590 DATA ED,B0,21,04,01,CD,75,BB, 3E,14,CD [3230]
,5A,BB,16,20,CD
600 DATA 9B,BC,21,20,8C,CD,D9,88, 3E,0E,CD [1583]
,39,83,E5,21,00
610 DATA 40,11,00,C0,44,4D,ED,B0, CD,6A,88 [3357]
,E1,E3,CD,75,BB
620 DATA E1,46,23,C9,B7,ED,52,D8, 7C,FE,40 [2497]
,3F,C9,3A,8C,89
630 DATA B7,C0,3A,8B,89,B7,C8,37, C9,32,8A [2248]
,89,21,7B,89,54
640 DATA 5D,13,97,77,32,8B,89,CD, 52,83,CD [3762]
,84,BB,7E,32,8C
650 DATA 89,C9,CD,81,BB,CD,06,BB, FE,0D,C8 [2724]
,E5,CD,6A,83,E1
660 DATA 3E,14,32,8B,89,CD,5A,BB, 18,E8,FE [3413]
,7F,28,38,FE,FC
670 DATA CA,2D,89,FE,21,D8,FE,30, 38,17,FE [1934]
,3A,38,1B,FE,41
680 DATA 38,0F,47,E6,DF,FE,47,38, 10,FE,5B [2691]
,38,04,78,FE,7F
690 DATA D0,47,3A,8A,89,FE,0A,D8, 78,47,3A [2548]
,8A,89,BE,78,C8
700 DATA 34,12,13,C3,5A,BB,97,BE, C8,35,1B [2642]
,3E,08,C3,5A,BB
710 DATA 21,78,8C,CD,D9,88,DF,BA, 83,C9,BD [3048]
,83,FF,DD,21,6D
720 DATA 89,ED,4B,73,89,CB,B0,03, 2A,6D,89 [2105]
,11,00,40,DD,7E
730 DATA 01,BA,38,10,20,05,DD,7E, 00,B7,C8 [877]
,DD,66,01,DD,6E
740 DATA 00,ED,B0,C9,DD,7E,03,BA, 38,F1,67 [2256]
,DD,6E,02,50,59
750 DATA CB,F2,1B,ED,B8,C9,21,3C, 8C,CD,D9 [2452]
,88,21,01,00,22
760 DATA 79,89,21,44,00,22,8E,89, 11,00,40 [3002]
,7B,E6,FC,CC,5D
770 DATA 84,1A,CD,6B,84,CD,26,89, 2A,73,89 [2492]
,B7,ED,52,30,EB
780 DATA 1B,7B,CD,5D,84,06,08,3E, 09,CD,4F [2855]
,88,2A,79,89,2B
790 DATA E5,3E,26,CD,5A,BB,CB,3C, CB,1D,CB [2779]
,3C,CB,1D,22,73
800 DATA 89,CD,F9,88,21,1F,8D,06, 0C,7B,3D [2287]
,B2,20,01,05,7E
810 DATA CD,5A,BB,23,10,F9,E1,CB, F4,22,79 [2838]
,89,C9,D5,EB,6F
820 DATA CD,F3,88,CD,F9,88,CD,4B, 88,D1,C9 [1967]
,13,47,CB,98,4F
830 DATA FE,40,38,40,FE,C0,D8,E6, 07,C8,FE [1343]
,06,28,5D,79,FE
840 DATA CB,28,58,FE,ED,CA,1A,85, FE,C3,28 [3884]
,1B,FE,CD,28,17
850 DATA E6,DF,FE,DD,28,47,E6,D7, FE,D3,28 [2319]
,3F,E6,C7,FE,C7
860 DATA 28,05,E6,C1,FE,C1,C8,1B, F5,CD,22 [3030]
,85,F1,13,3A,8D
870 DATA 89,C3,45,88,E6,CF,3D,28, 21,FE,08 [3045]
,C8,78,E6,CF,FE
880 DATA 06,28,18,78,E6,EF,FE,22, 28,10,78 [2643]
,B7,C8,E6,CF,C0
890 DATA 21,9B,89,CB,46,20,D0,18, 02,13,13 [3191]
,13,C9,1A,47,CB
900 DATA 98,4F,FE,E0,30,F5,FE,CB, 28,EF,47 [2278]
,FE,23,28,EC,79
910 DATA E6,CF,FE,09,28,E5,79,FE, 36,28,DE [3990]
,3D,E6,FE,FE,20
```

Listing SOLVE

```
920 DATA 28,D7,78,E6,FE,FE,34,28, D1,C6,40 [1515]
,E6,8F,FE,86,28
930 DATA C9,79,E6,F8,FE,70,28,C2, 18,C1,E6 [2452]
,78,FE,40,28,B9
940 DATA 18,B9,D5,CD,B7,85,ED,5B, 69,89,CD [2978]
,5D,88,38,58,ED
950 DATA 5B,6B,89,CD,5D,88,38,02, 20,4D,E5 [2657]
,ED,4B,79,89,5D
960 DATA 21,FF,3F,CD,6D,88,7B,ED, B1,E2,89 [2575]
,85,7D,E6,03,3D
970 DATA 20,F4,D1,7A,BE,D5,20,EE, D1,23,5E [3048]
,23,56,EB,0E,02
980 DATA CD,71,88,7E,23,B6,2B,28, 12,0D,23 [3237]
,23,20,F5,7E,23
990 DATA B6,2B,C4,63,88,20,EC,CD, 9E,85,EB [1573]
,D1,73,23,72,97
1000 DATA 23,77,23,77,C3,6A,88,D1, C9,D1,7 [2473]
3,23,72,23,CD,9E
1010 DATA 85,23,CB,B4,23,22,79,89, EB,CD,7 [2874]
1,88,18,DD,ED,5B
1020 DATA 8E,89,7A,FE,80,D2,76,88, 73,23,7 [2481]
2,D5,3E,06,CD,45
1030 DATA 88,ED,53,8E,89,D1,C9,21, 8D,89,3 [2741]
6,00,79,E6,C7,FE
1040 DATA C7,28,23,FE,C0,13,1A,13, 34,38,0 [2137]
5,34,6F,1A,67,C9
1050 DATA FE,80,38,0B,EB,16,00,ED, 44,5F,B [3703]
7,ED,52,18,04,CD
1060 DATA 45,88,EB,C3,F3,88,79,E6, 38,6F,2 [2942]
6,00,C9,ED,5B,79
1070 DATA 89,CD,F1,86,3E,3F,BA,D0, 21,4F,8 [2642]
C,CD,D9,88,CD,6D
1080 DATA 88,3E,02,32,96,89,21,00, 00,22,9 [1807]
7,89,CB,F4,D5,E5
1090 DATA 21,96,89,35,CA,6A,88,CD, 26,89,E [3538]
1,22,92,89,D1,ED
1100 DATA 53,94,89,CD,72,86,CD,84, 86,D5,E [2379]
5,CD,A8,86,30,19
1110 DATA ED,5B,92,89,E1,E3,EB,CD, 5D,88,C [3012]
1,30,06,D5,E5,21
1120 DATA 96,89,34,ED,43,92,89,18, 15,ED,5 [2189]
B,94,89,E1,CD,5D
1130 DATA 88,C1,30,06,D5,E5,21,96, 89,34,E [2649]
D,43,94,89,CC,5C
1140 DATA 89,2A,92,89,ED,5B,94,89, CD,5D,8 [1807]
8,38,B6,CC,5C,89
1150 DATA 18,9E,EB,E5,B7,ED,52,CB, 2C,CB,1 [3273]
D,CB,8D,19,4E,23
1160 DATA 46,E1,EB,C9,CD,BC,86,CD, CB,86,C [3290]
D,5D,88,38,0A,C0
1170 DATA E5,D5,C5,CD,5C,89,C1,D1, E1,CD,D [2777]
A,86,CD,EC,86,CD
1180 DATA 5D,88,CD,F1,86,38,DD,C9, E5,2A,9 [2192]
4,89,ED,5B,92,89
1190 DATA B7,ED,52,CB,2C,CB,1D,19, D1,C3,5 [2236]
D,88,23,78,BE,2B
1200 DATA 20,02,79,BE,D8,C8,CD,EC, 86,18,F [2712]
1,13,1A,B8,1B,20
1210 DATA 02,1A,B9,D8,C8,CD,F1,86, 18,F1,C [2410]
5,E5,D5,06,04,1A
1220 DATA 4E,77,79,12,23,13,10,F7, D1,E1,C [2544]
1,C9,23,23,23,23
1230 DATA C9,1B,1B,1B,1B,C9,21,64, 8C,CD,D [2169]
9,88,CD,31,88,21
1240 DATA 9C,89,23,CD,34,88,21,96, 8C,CD,3 [2396]
4,88,3E,26,CD,0A
1250 DATA 88,2A,6D,89,CD,19,88,21, 52,8A,C [1864]
D,34,88,2A,6F,89
1260 DATA CD,19,88,CD,31,88,06,06, 3E,20,C [2625]
D,55,88,3E,26,CD
1270 DATA 0A,88,2A,73,89,CD,19,88, 21,1F,8 [2855]
D,CD,34,88,2A,69
1280 DATA 89,CD,19,88,21,52,8A,CD, 34,88,2 [2559]
A,6B,89,CD,19,88
1290 DATA 3E,20,CD,0A,88,3E,28,CD, 0A,88,3 [2185]
A,9B,89,21,FC,8C
1300 DATA CB,EE,06,14,B7,20,0C,21, F5,8C,C [3415]
D,34,88,CD,EC,86
1310 DATA 23,06,10,CD,3D,88,3E,29, CD,0A,8 [3285]
8,CD,31,88,CD,31
1320 DATA 88,3E,26,CD,5A,BB,21,00, 00,22,9 [2732]
7,89,CB,F4,ED,5B
1330 DATA 79,89,CD,5D,88,D2,6A,88, CD,6D,8 [3335]
8,E5,CD,65,88,CD
1340 DATA 19,88,3E,3A,CD,0A,88,CD, 5C,89,E [2398]
1,23,23,E5,CD,63
1350 DATA 88,CD,71,88,06,00,CD,BE, 87,E1,2 [2159]
3,23,18,D0,E5,CD
```

Listing SOLVE

```
1360 DATA 65,88,7C,B5,28,11,CD,E0, 87,E1,2 [2629]
3,23,0D,20,EF,CD
1370 DATA 63,88,7C,B5,20,E8,E5,E1, 3E,2E,C [3280]
D,0A,88,C3,31,88
1380 DATA 3E,20,CD,0A,88,78,FE,07, 3E,20,C [1840]
C,0A,88,78,FE,0E
1390 DATA CC,FB,87,CD,F3,88,CD,19, 88,04,C [2728]
9,E5,CD,31,88,06
1400 DATA 06,3E,20,CD,55,88,06,00, E1,C9,E [2765]
5,C5,F5,CD,26,89
1410 DATA F1,C1,E1,CD,2B,BD,D8,18, F1,E5,C [3079]
5,11,7B,89,CD,03
1420 DATA 89,EB,CD,34,88,C1,E1,C9, E5,21,0 [2026]
5,8B,CD,D9,88,E1
1430 DATA C9,21,05,8B,7E,B7,23,C8, CD,0A,8 [1870]
8,18,F7,7E,CD,0A
1440 DATA 88,23,10,F9,C9,83,5F,8A, 93,57,C [2326]
9,06,04,3E,08,CD
1450 DATA 5A,BB,10,FB,C9,F5,CD,0A, 88,F1,1 [3344]
0,F9,C9,7C,BA,C0
1460 DATA 7D,BB,C9,0E,02,7E,23,66, 6F,C9,9 [3350]
7,18,06,3E,04,18
1470 DATA 02,3E,05,C3,39,8D,21,F5, 89,E5,2 [3678]
1,CC,89,CD,D9,88' &8d39 RAM-KONF
1480 DATA E1,CD,D9,88,21,DA,89,CD, D9,88,C [2632]
D,06,BB,3E,14,C3
1490 DATA 5A,BB,DD,21,BE,88,21,00, 00,B7,C [2276]
8,D5,FD,E1,11,01
1500 DATA 00,47,FD,2B,FD,7E,00,CD, C4,88,D [2605]
8,28,04,19,3D,20
1510 DATA FC,DD,5E,00,DD,56,01,DD, 23,DD,2 [3244]
3,10,E5,C9,10,00
1520 DATA 00,01,00,10,FE,41,38,07, FE,5B,3 [2499]
F,D8,D6,37,C9,FE
1530 DATA 30,D8,FE,3A,3F,D8,D6,30, C9,7E,2 [2666]
3,B7,C8,FE,09,CC
1540 DATA E7,88,CD,5A,BB,18,F2,C5, 06,0C,3 [2431]
E,20,CD,5A,BB,10
1550 DATA F9,C1,C9,ED,5B,90,89,19, C9,11,7 [3917]
B,89,CD,03,89,EB
1560 DATA C3,D9,88,D5,7C,CD,10,89, 7D,CD,1 [3072]
0,89,97,12,D1,C9
1570 DATA F5,0F,0F,0F,0F,CD,19,89, F1,E6,0 [2973]
F,FE,0A,38,02,C6
1580 DATA 07,C6,30,12,13,C9,CD,09, BB,D0,F [2082]
E,FC,C0,3E,07,CD
1590 DATA 5A,BB,CD,78,BB,E5,26,01, 2C,CD,7 [3387]
5,BB,21,C1,8C,CD
1600 DATA D9,88,CD,81,BB,CD,3D,BD, CD,06,B [2711]
B,FE,FC,CA,12,80
1610 DATA CD,84,BB,3E,11,CD,5A,BB, E1,C3,7 [2385]
5,BB,2A,97,89,23
1620 DATA 22,97,89,CD,F9,88,C3,4B, 88,00,0 [2369]
0,FF,FF,00,C0,FF
1630 DATA FF,00,00,00,40,69,89,57, 8A,00,0 [1641]
0,62,79,74,E5,EF
1640 DATA FF,1D,28,0C,00,00,6F,B1, 29,01,0 [2625]
4,00,00,00,00,00
1650 DATA 00,00,00,40,00,00,00,00, 40,00,0 [2907]
0,00,09,53,4F,4C
1660 DATA 56,45,20,31,2E,39,20,20, 20,20,2 [2202]
0,20,31,31,2E,38
1670 DATA 2E,31,39,38,38,20,62,79, 20,48,6 [2669]
5,6C,6D,75,74,20
1680 DATA 53,74,65,65,62,09,53,44, 47,0D,0 [2363]
A,00,20,2A,2A,2A
1690 DATA 2A,20,45,52,52,4F,52,3A, 20,00,2 [2689]
0,2D,2D,2D,20,57
1700 DATA 65,69,74,65,72,20,6D,69, 74,20,2 [2399]
8,54,41,53,54,45
1710 DATA 29,20,8F,0D,00,41,75,66, 72,75,6 [2226]
6,2D,4C,69,73,74
1720 DATA 65,20,76,6F,6C,6C,21,00, 42,69,7 [2833]
4,74,65,20,45,6E
1730 DATA 64,61,64,72,65,73,73,65, 20,2D,2 [2689]
0,53,74,61,72,74
1740 DATA 61,64,72,65,73,73,65,20, 3C,20,2 [2286]
6,34,30,30,30,20
1750 DATA 21,00,4E,61,6D,65,20,75, 6E,7A,7 [2422]
5,6C,61,65,73,73
1760 DATA 69,67,00,44,61,74,65,69, 20,7A,7 [3365]
5,20,4C,41,4E,47
1770 DATA 21,00,20,2D,20,26,00,0D, 09,41,7 [1575]
5,66,72,75,66,65
1780 DATA 20,69,6E,20,42,65,72,65, 69,63,6 [3082]
8,20,61,62,20,2E
1790 DATA 2E,2E,2E,2E,2E,2E,20,26, 30,30,3 [2565]
0,30,00,0D,09,20
```

Listing SOLVE

```
1800 DATA 20,20,20,20,20,20,20,20, 20,20,2 [1961]
0,20,20,20,20,20
1810 DATA 20,20,62,69,73,20,2E,2E, 2E,2E,2 [2563]
E,2E,20,26,46,46
1820 DATA 46,46,00,0D,09,55,6E,74, 65,72,7 [2651]
3,75,63,68,75,6E
1830 DATA 67,20,53,74,61,72,74,61, 64,72,6 [2535]
5,73,73,65,20,2E
1840 DATA 2E,2E,20,26,43,30,30,30, 00,0D,0 [3524]
9,20,20,20,20,20
1850 DATA 20,20,20,20,20,20,20,20, 45,6E,6 [2216]
4,61,64,72,65,73
1860 DATA 73,65,20,2E,2E,2E,2E,2E, 20,26,4 [2849]
6,46,46,46,00,20
1870 DATA 09,09,09,09,41,62,62,72, 75,63,6 [2609]
8,20,6D,69,74,20
1880 DATA 28,45,53,43,29,0D,0A,00, 0A,09,0 [2099]
9,20,55,4E,54,45
1890 DATA 52,53,55,43,48,55,4E,47, 3A,0D,0 [1941]
A,09,09,D0,D0,D0
1900 DATA D0,D0,D0,D0,D0,D0,D0,D0, D0,D0,D [1937]
0,D0,0D,0A,09,42
1910 DATA 65,74,72,69,65,62,73,73, 79,73,7 [3157]
4,65,6D,20,28,4C
1920 DATA 6F,77,65,72,20,52,4F,4D, 29,20,2 [2832]
E,2E,20,30,0D,0A
1930 DATA 09,42,41,53,49,43,20,28, 55,70,7 [2826]
0,65,72,20,52,4F
1940 DATA 4D,29,20,2E,2E,2E,2E,2E, 2E,2E,2 [3818]
E,2E,2E,2E,20,31
1950 DATA 0D,0A,09,52,41,4D,2D,42, 65,72,6 [2225]
5,69,63,68,20,28
1960 DATA 6D,61,78,2E,20,31,36,6B, 29,20,2 [2583]
E,2E,2E,2E,2E,2E
1970 DATA 20,32,0D,0A,09,45,78,70, 61,6E,7 [1513]
3,69,6F,6E,2D,52
1980 DATA 4F,4D,20,2E,2E,2E,2E,2E, 2E,2E,2 [3749]
E,2E,2E,2E,2E,2E
1990 DATA 2E,2E,20,33,0D,0A,00,0D, 09,4D,4 [3212]
3,2D,44,41,54,45
2000 DATA 49,20,28,6D,61,78,2E,20, 31,36,6 [1496]
B,29,20,2E,2E,2E
2010 DATA 2E,2E,2E,2E,2E,2E,20,34, 20,20,2 [2952]
0,20,20,20,3F,20
2020 DATA 34,20,20,20,00,0D,09,4E, 75,6D,6 [2684]
D,65,72,20,64,65
2030 DATA 73,20,52,4F,4D,20,28,44, 69,73,6 [2790]
B,3D,30,37,29,20
2040 DATA 2E,2E,2E,2E,20,26,30,37, 20,20,0 [2925]
0,0A,09,44,69,65
2050 DATA 20,44,61,74,65,69,20,62, 65,6C,6 [2513]
5,67,74,20,26,00
2060 DATA 09,44,61,74,65,69,6E,61, 6D,65,2 [2377]
0,28,46,49,4C,45
2070 DATA 4E,41,4D,45,2E,45,58,54, 29,3A,2 [1745]
0,00,55,4E,54,45
2080 DATA 52,53,55,43,48,55,4E,47, 2E,2E,2 [2311]
E,20,20,26,00,0D
2090 DATA 0A,53,4F,52,54,49,45,52, 45,4E,2 [2671]
E,2E,2E,20,20,20
2100 DATA 20,20,26,00,0D,0A,41,55, 53,47,4 [1889]
1,42,45,2E,2E,2E
2110 DATA 20,20,20,20,20,20,20,00, 0D,0A,5 [2504]
6,45,52,53,43,48
2120 DATA 49,45,42,45,4E,20,6E,61, 63,68,2 [2122]
0,26,34,30,30,30
2130 DATA 2E,2E,2E,0D,0A,00,55,6E, 74,65,7 [3303]
2,73,75,63,68,75
2140 DATA 6E,67,3A,20,20,84,01,C9, 00,0C,0 [1871]
0,26,84,0B,00,00
2150 DATA E2,EF,F5,0F,00,1B,00,30, 84,9E,2 [3453]
0,0B,00,00,E1,EF
2160 DATA 0E,20,2A,2A,2A,2A,20,53, 4F,4C,5 [2419]
6,45,20,62,65,65
2170 DATA 6E,64,65,6E,20,6D,69,74, 20,28,4 [2661]
5,53,43,29,20,2F
2180 DATA 20,57,65,69,74,65,72,20, 6D,69,7 [2471]
4,20,28,54,61,73
2190 DATA 74,65,29,20,00,6F,68,6E, 65,00,0 [1787]
D,09,4D,69,74,20
2200 DATA 52,65,6C,61,74,69,76,73, 70,72,7 [2295]
5,65,6E,67,65,6E
2210 DATA 20,28,30,2F,31,29,3F,20, 2E,20,3 [2270]
0,20,20,20,00,20
2220 DATA 53,70,72,75,6E,67,7A,69, 65,6C,6 [2174]
5,20,69,6D,20,42
2230 DATA 65,72,65,69,63,68,20,26, 00,F3,D [2328]
9,F6,C0,ED,79,D9
2240 DATA FB,C9,00                       [432]
```

Listing SOLVE



MICRO-PAINT

Grafik mit Pfiff

Micro Paint ist ein Zeichenprogramm, welches durch seine Möglichkeiten überzeugt. Es ist eine verbesserte Version des in Heft 7/89 abgedruckten Programmes DRAW.

Die mit diesem Programm erstellten Grafiken können mit vielen anderen Programmen, wie zum Beispiel ART-WORX, weiterverarbeitet werden.

Am Anfang

Im Ladeprogramm werden Sie zuerst nach dem CPC-Typ gefragt. Wenn Sie einen CPC464 benutzen, verlangt das Programm den Emulator, welcher sich auch in diesem Heft befindet.

Ist dieser geladen, oder haben Sie bei der Typ-Abfrage mit 664 beziehungsweise 6128 geantwortet, fragt der Computer, ob Sie das Programm mm-copy aus Heft 10/88 laden wollen. Wenn ja, dann sollte es sich auf der gleichen Diskette/Kassette wie das MICRO-PAINT-Hauptprogramm befinden.

Nun wird noch eine Meldung ausgegeben, welche angibt, ob die Funktion Drucken benutzt werden kann oder nicht.

Als nächstes wird das eigentliche Programm geladen, und Sie befinden sich im Hauptmenü. Hier können die einzelnen Funktionen durch die Tasten 'Hoch' und 'Runter' angewählt werden. Bestätigt wird nun mit COPY.

Folgende Funktionen werden vom Programm zur Verfügung gestellt.

Bleistift
Mit COPY wird an der aktuellen Position in der jeweiligen Stiftfarbe ein Punkt gesetzt.

Lineal
Nachdem Sie mit COPY den Anfangspunkt gesetzt haben, müssen Sie den Endpunkt anfahren, und eine Linie wird gezeichnet.

Füllen
Setzen Sie den Cursor auf die auszufüllende Fläche und drücken Sie COPY. Die Fläche wird nun in der aktuellen Stiftfarbe ausgefüllt.

Spraydose dünn
Durch drücken von COPY wird dünn gesprüht.

Spraydose dick
Wie oben, es werden jedoch keine einzelnen Punkte gesetzt, sondern ganze 8*8 Pixel große Quadrate.

Platten legen
Hier können Platten gelegt werden, die zuvor definiert wurden (siehe 'Platten definieren').

Platten definieren
Nach zirka 20 Sekunden wird die aktuelle Platte rechts vergrößert dargestellt, in der Mitte in Originalgröße gezeichnet und links erscheint ein Cursor. Nun können Sie Ziffern von null bis drei, gefolgt von ENTER, eingegeben. Diese Ziffern werden dann in Farben umgewandelt (0=Schwarz, 1=Gelb, 2=Hellblau, 3=Rot) und rechts in der Vergrößerung gezeichnet. Das Original in der Mitte wird auch verändert, damit Sie gleich sehen, wie die Platte auf dem Bildschirm aussehen wird. Wenn Sie die ganze 8*8 Matrix in Ziffern eingegeben haben, erscheint wieder das Hauptmenü, und über die Funktion 'Platten legen' kann die definierte Platte beliebig oft abgelegt werden.

Screen manipulieren
Hier kann das Bild mit einem Wert von 0-255 manipuliert werden. Die Zahl Null hat jedoch keine Auswirkung.

Wenn Sie den Screen manipuliert haben, und diese Funktion rückgängig machen wollen, weil Ihnen das Ergebnis nicht gefällt, manipulieren Sie mit der gleichen Zahl nochmal. (Lassen Sie sich überraschen, was passiert.)

Zirkel
Eigentlich ist diese Funktion falsch benannt, weil nicht nur Kreise, sondern auch Ellipsen gezeichnet werden können: Mittelpunkt anfahren, COPY drücken, danach rechten beziehungsweise linken Punkt der Ellipse anfahren, wieder COPY drücken, nun wird ein senkrechter Strich gezogen, und der Grafikcursor erscheint oberhalb beziehungsweise unterhalb des Mittelpunktes. Drücken Sie nochmals COPY, so wird ein Kreis gezeichnet. Sollten Sie jedoch hoch oder runter fahren, können Sie auch Ellipsen zeichnen.

PEN-Wechsel
Hier kann die neue Stiftfarbe eingestellt werden (0-3).

Text
Bewegen Sie den Cursor an die Position, an welcher der Text stehen soll, drücken Sie COPY und geben Sie den Text ein. Die Eingabe kann nun mit ENTER beendet werden.

Großtext
Hier geben Sie zuerst den Text und dann den Vergrößerungsfaktor ein. Als nächstes bewegen Sie den Cursor an die Position, an welcher der Text plaziert werden soll und drücken COPY.

Laden/Speichern
17-kByte-Bilder können hier entweder geladen oder abgespeichert werden.

Symbole setzen
Hier kann eines der 17 einprogrammierten Symbole gezeichnet werden: Symbolnummer eingeben, Symbol wird auf die Mitte des Bildschirms gezeichnet. Nun wird der Arbeitsscreen eingeblendet und das Symbol kann mittels COPY an

an beliebige Positionen gesetzt werden. Nun wird das Symbol gezeichnet und auf eine Bestätigung gewartet, das bedeutet, wenn Sie 'S' (für setzen) drücken, wird das Symbol endgültig gesetzt, drücken Sie aber 'Z' (für zurücknehmen), so wird das Symbol wieder gelöscht.

MICRO-PAINT kennt folgende Symbole: Joystick, Computer, Monitor, Taschenrechner, Diskette, Buch, Schneider-Zeichen, CPC-Zeichen, zwei verschiedene Ansichten eines Diskettenlaufwerks, drei verschiedene Wolken, zwei verschiedene Stühle, Tisch und ein Haus.

Drucken

Diese Funktion läuft nur, wenn vom Ladeprogramm aus das Programm 'mm-copy.bin' geladen wurde. Zur Sicherheit wird dies noch einmal abgefragt. Nun können Sie die Größe des Bildes bestimmen.

Jetzt erscheint eine weitere Sicherheitsabfrage. Bestätigen Sie diese, obwohl 'mm-copy.bin' nicht geladen wurde, stürzt das Programm ab und das Bild geht verloren.

Löschen

Nach einer Sicherheitsabfrage wird das Bild gelöscht.

(Christoph Laube/rs)

für 464-664-6128

```
10 PAPER 0:PEN 1:INK 0,0:INK 1,26:MODE 2:B [3960]
ORDER 0
20 MEMORY &5FFF                             [102]
30 INPUT"Benutzen Sie einen CPC 464/664/61 [3026]
28 ?";a
40 IF a<>464 AND a<>664 AND a<>6128 THEN P [3785]
RINT CHR$(7):GOTO 10
50 IF a<>464 THEN 110                      [1299]
60 PRINT"Bitte legen Sie eine Kassette/Dis [9357]
kette mit dem 664-6128 Emulator ein und
     druecken Sie eine Taste !"
70 WHILE INKEY$<>"":WEND                   [1786]
80 WHILE INKEY$="":WEND                    [1607]
90 LOAD "emu.bin",&A200                    [2257]
100 CALL &A200                             [597]
110 CLS:PRINT"MM-COPY.BIN laden (J/N) ?"   [3132]
120 a$=LOWER$(INKEY$)                      [960]
130 IF a$="n" THEN CLS:PRINT"Die Funktion  [7257]
Drucken kann nicht benutzt werden !!!":GOT
O 210
140 IF a$<>"j" THEN 120                    [1359]
150 PRINT"Bitte legen Sie eine Kassette/Di [8749]
skette mit mm-copy.bin ein und druecken Si
e eineTaste !"
160 WHILE INKEY$<>"":WEND                  [1786]
170 WHILE INKEY$="":WEND                   [1607]
180 CLS                                    [91]
190 LOAD "mm-copy.bin",&A000               [1951]
200 PRINT"Die Funktion Ausdrucken kann nun [4926]
  benutzt werden !"
210 RUN "paint.hpt"                        [759]
```

```
10 '***************************            [1051]
20 '*          MICRO-PAINT         *       [1216]
30 '*                              *       [175]
40 '*     written by C.Lauble      *       [1533]
50 '*                              *       [175]
60 '*          in 1/1989           *       [298]
70 '***************************            [1051]
80 '                                       [117]
90 '                                       [117]
100 f=1:DIM pl(8,8):DEG                    [703]
110 IF PEEK(&A000)<>0 THEN CALL &A000      [2615]
120 FOR x=&A100 TO &A16C                   [1335]
130   READ a$:b=VAL("&"+a$)                [779]
140   s=s+b:POKE x,b                       [494]
150 NEXT x                                 [356]
160 IF s<>10947 THEN PRINT"DATA-Fehler !": [1961]
END
170 DATA 0E,A1,C3,29,A1,C3,41,A1           [1484]
180 DATA C3,4D,A1,C3,59,A1,53,43           [1620]
190 DATA D2,49,CE,4F,55,D4,43,4C           [1395]
200 DATA D3,00,00,00,00,00,00,00           [1182]
210 DATA 01,00,A1,21,1A,A1,C3,D1           [1802]
220 DATA BC,DD,46,00,21,00,C0,11           [1465]
230 DATA 00,40,7E,A8,77,23,1B,E5           [979]
240 DATA 21,FF,FF,19,E1,DA,32,A1           [1400]
250 DATA C9,21,00,60,11,00,C0,01           [1081]
260 DATA 00,40,ED,B0,C9,21,00,C0           [1088]
270 DATA 11,00,60,01,00,40,ED,B0           [883]
280 DATA C9,21,00,60,01,00,40,36           [1440]
290 DATA 00,23,0B,E5,21,FF,FF,09           [1552]
300 DATA E1,DA,5F,A1,C9                    [728]
310 CALL &A120                             [629]
320 '                                      [117]
330 '>>> MENUE <<<                          [463]
340 '                                      [117]
350 MODE 1:INK 1,24                        [1198]
360 LOCATE 15,3:PRINT"MICRO-PAINT"         [1826]
```

Listing MICRO PAINT

```
370 LOCATE 8,5:PRINT"fuer 464+Emulator,664 [2720]
,6128"
380 RESTORE 420                            [775]
390 FOR x=1 TO 17                          [820]
400    READ a$:LOCATE 6,7+x:PRINT a$       [1886]
410 NEXT                                   [350]
420 DATA Bleistift,Lineal,Fuellen,Spraydos [5932]
e duenn,Spraydose dick,Platten legen
430 DATA Platten definieren,Screen Manipul [4449]
ieren,Zirkel,PEN-wechsel
440 DATA Text,Grosstext,Laden,Speichern,Sy [5917]
mbole setzten,Drucken,Loeschen
450 DATA PEN-wechsel                       [933]
460 LOCATE 4,8:PRINT CHR$(243):y=1         [2865]
470 IF INKEY(72)=0 OR INKEY(0)=0 THEN 560  [1813]
480 IF INKEY(73)=0 OR INKEY(2)=0 THEN 510  [2034]
490 IF INKEY(77)=0 OR INKEY(76)=0 OR INKEY [3261]
(9)=0 THEN 610
500 GOTO 470                               [454]
510 IF y=17 THEN LOCATE 4,24:PRINT" ":y=0: [2296]
GOTO 480
520 LOCATE 4,7+y:PRINT" ":y=y+1            [2784]
530 LOCATE 4,7+y:PRINT CHR$(243)           [2944]
540 FOR x=1 TO 100:NEXT                    [1450]
550 GOTO 480                               [373]
560 IF y=1 THEN LOCATE 4,8:PRINT" ":y=18:G [2973]
OTO 490
570 LOCATE 4,7+y:PRINT" ":y=y-1            [2780]
580 LOCATE 4,7+y:PRINT CHR$(243)           [2944]
590 FOR x=1 TO 100:NEXT                    [1450]
600 GOTO 490                               [371]
610 MODE 1:WHILE INKEY$<>"":WEND           [2586]
620 ON y GOTO 890,960,1090,1170,1250,1330, [4613]
1450,1730,1830,2060,2120,2240,2550,2780,29
30,4100,4370
630 '                                      [117]
640 '>>> Cursor bewegen <<<                [1008]
650 '                                      [117]
660 PLOT xp,yp,f1                          [1383]
670 IF INKEY(0)=0 OR INKEY(72)=0 THEN yp=y [3233]
p+1
680 IF INKEY(2)=0 OR INKEY(73)=0 THEN yp=y [1663]
p-1
690 IF INKEY(8)=0 OR INKEY(74)=0 THEN xp=x [2139]
p-1
700 IF INKEY(1)=0 OR INKEY(75)=0 THEN xp=x [2314]
p+1
710 IF INKEY(0)=32 OR INKEY(72)=32 THEN yp [2021]
=yp+8
720 IF INKEY(2)=32 OR INKEY(73)=32 THEN yp [2300]
=yp-8
730 IF INKEY(8)=32 OR INKEY(74)=32 THEN xp [1500]
=xp-8
740 IF INKEY(1)=32 OR INKEY(75)=32 THEN xp [1980]
=xp+8
750 IF INKEY(0)=128 OR INKEY(72)=128 THEN  [1923]
yp=yp+32
760 IF INKEY(2)=128 OR INKEY(73)=128 THEN  [1619]
yp=yp-32
770 IF INKEY(8)=128 OR INKEY(74)=128 THEN  [2496]
xp=xp-32
780 IF INKEY(1)=128 OR INKEY(75)=128 THEN  [2723]
xp=xp+32
790 IF INKEY(9)=0 OR INKEY(76)=0 OR INKEY( [2349]
77)=0 THEN 840
800 IF INKEY(38)=0 THEN PLOT xp,yp,f1:¦OUT [1590]
:GOTO 350
810 f1=TEST(xp,yp):PLOT xp,yp,f1+2         [975]
820 FOR x=1 TO 50:NEXT                     [1564]
830 GOTO 660                               [369]
840 WHILE INKEY$<>"":WEND                  [1786]
850 RETURN                                 [555]
860 '                                      [117]
870 '>>> Bleistift <<<                     [1013]
880 '                                      [117]
890 ¦IN:xp=0:yp=0:f1=TEST(xp,yp)           [1495]
```

Listing MICRO PAINT

```
900 GOSUB 660                                       [881]
910 PLOT xp,yp,f                                    [1322]
920 f1=f:GOTO 900                                   [509]
930 '                                               [117]
940 '>>> Lineal <<<                                 [474]
950 '                                               [117]
960 |IN:xp=0:yp=0:f1=TEST(xp,yp)                    [1495]
970 x1=0:y1=0                                       [1050]
980 GOSUB 660                                       [881]
990 PLOT xp,yp,f:x1=xp:y1=yp:f1=f                   [2764]
1000 FOR x=1 TO 100:NEXT                            [1450]
1010 WHILE INKEY$<>"":WEND                          [1786]
1020 GOSUB 660                                      [881]
1030 MOVE x1,y1:DRAW xp,yp,f                        [540]
1040 f1=f                                           [99]
1050 GOTO 970                                       [330]
1060 '                                              [117]
1070 '>>> Fuellen <<<                               [1248]
1080 '                                              [117]
1090 |IN:xp=0:yp=0:f1=TEST(xp,yp)                   [1495]
1100 GOSUB 660                                      [881]
1110 PLOT xp,yp,f:|OUT:MODE 1:MOVE xp,yp+2 [3819]
:|IN:FILL f
1120 f1=f                                           [99]
1130 GOTO 1100                                      [351]
1140 '                                              [117]
1150 '>>> Spraydose duenn <<<                       [1294]
1160 '                                              [117]
1170 |IN:xp=0:yp=0:f1=TEST(xp,yp)                   [1495]
1180 GOSUB 660                                      [881]
1190 PLOT xp+RND*8,yp+RND*8,f                       [1577]
1200 f1=TEST(xp,yp)                                 [584]
1210 GOTO 1180                                      [399]
1220 '                                              [117]
1230 '>>> Spraydose dick <<<                        [763]
1240 '                                              [117]
1250 |IN:xp=0:yp=0:f1=TEST(xp,yp)                   [1495]
1260 GOSUB 660                                      [881]
1270 GRAPHICS PEN f:TAG:PRINT CHR$(143);:T [1871]
AGOFF
```

Listing MICRO PAINT

```
1280 f1=f                                           [99]
1290 GOTO 1260                                      [351]
1300 '                                              [117]
1310 '>>> Platten legen <<<                         [1574]
1320 '                                              [117]
1330 |IN:xp=0:yp=0:f1=TEST(xp,yp)                   [1495]
1340 GOSUB 660                                      [881]
1350 FOR x=1 TO 8                                   [706]
1360   FOR y=1 TO 8                                 [1182]
1370     PLOT xp+(x-1)*2,yp+(y-1)*2,pl(x,y [1570]
)
1380   NEXT y                                       [359]
1390 NEXT x                                         [356]
1400 f1=TEST(xp,yp)                                 [584]
1410 GOTO 1340                                      [462]
1420 '                                              [117]
1430 '>>> Platten definieren <<<                    [2284]
1440 '                                              [117]
1450 WINDOW 30,40,1,25                              [1193]
1460 FOR y=1 TO 8                                   [1182]
1470   FOR x=1 TO 8                                 [706]
1480     PEN pl(x,y)                                [732]
1490     LOCATE x,9-y                               [1019]
1500     PRINT CHR$(143)                            [856]
1510     PLOT 314+x*2,350+y*2,pl(x,y)               [2075]
1520   NEXT x                                       [356]
1530 NEXT y                                         [359]
1540 WINDOW 1,40,1,25                               [1024]
1550 PEN 1                                          [549]
1560 FOR y=1 TO 8                                   [1182]
1570   FOR x=1 TO 8                                 [706]
1580     LOCATE x,y                                 [672]
1590     LINE INPUT a$                              [1242]
1600     a=VAL(a$)                                  [956]
1610     LOCATE x+29,y:PEN a:PRINT CHR$(14 [1963]
3)
1620     pl(x,9-y)=a                                [829]
1630     PLOT 314+x*2,368-y*2,a                     [1116]
1640     PEN 1                                      [549]
1650   NEXT x                                       [356]
```

Listing MICRO PAINT

```
1660 NEXT y                                          [359]
1670 CLS:PRINT"Platte definiert !"                   [2273]
1680 FOR x=1 TO 1000:NEXT                            [1282]
1690 GOTO 350                                        [470]
1700 '                                               [117]
1710 '>>> Bild Manipulieren <<<                      [1691]
1720 '                                               [117]
1730 INPUT "Mit welchem Wert ";w                     [2385]
1740 IF w<>INT(w) THEN PRINT"Nur ganze Zah           [4158]
len !":PRINT:GOTO 1730
1750 IF w<0 THEN PRINT"Nur positive Zahlen           [4504]
 (und null) !":PRINT:GOTO 1730
1760 IF w>255 THEN PRINT"Nur kleinere Zahl           [4082]
en als 256 !":PRINT:GOTO 1730
1770 MODE 1                                          [506]
1780 |IN:|SCR,w:|OUT                                 [1489]
1790 GOTO 350                                        [470]
1800 '                                               [117]
1810 '>>> Zirkel <<<                                 [698]
1820 '                                               [117]
1830 |IN:xp=0:yp=0:f1=TEST(xp,yp)                    [1495]
1840 GOSUB 660                                       [881]
1850 PLOT xp,yp,f:f1=f:mx=xp:my=yp                   [3527]
1860 FOR x=1 TO 100:NEXT                             [1450]
1870 WHILE INKEY$<>"":WEND                           [1786]
1880 GOSUB 660                                       [881]
1890 PLOT xp,0:DRAWR 0,400,f                         [1319]
1900 rx=ABS(xp-mx)                                   [436]
1910 FOR x=1 TO 100:NEXT                             [1450]
1920 WHILE INKEY$<>"":WEND                           [1786]
1930 xp=mx:yp=my+rx:GOSUB 660                        [1850]
1940 ry=ABS(yp-my)                                   [478]
1950 CLS:|IN                                         [333]
1960 PLOT mx,my+ry                                   [1067]
1970 FOR x=0 TO 360 STEP 5                           [870]
1980   DRAW mx+rx*SIN(x),my+ry*COS(x),f              [1785]
1990 NEXT                                            [350]
2000 f1=TEST(xp,yp):PLOT mx,my,f1                    [926]
2010 xp=mx:yp=my                                     [1328]
2020 |OUT:GOTO 1840                                  [792]
2030 '                                               [117]
2040 '>>> PEN-wechsel <<<                            [1331]
2050 '                                               [117]
2060 LINE INPUT "Neuer PEN : ";a$                    [2712]
2070 IF a$<>"1" AND a$<>"2" AND a$<>"3" AN           [3663]
D a$<>"0" THEN PRINT CHR$(7):PRINT:PRINT:G
OTO 2060
2080 f=VAL(a$)                                       [425]
2090 GOTO 350                                        [470]
2100 '>>> Text <<<                                   [395]
2110 '                                               [117]
2120 |IN:xp=0:yp=0:f1=TEST(xp,yp)                    [1495]
2130 GOSUB 660                                       [881]
2140 PLOT xp,yp,f1                                   [1383]
2150 GRAPHICS PEN f                                  [944]
2160 TAG                                             [318]
2170 a$=INKEY$                                       [278]
2180 IF a$=CHR$(13) THEN TAGOFF:GOTO 2130            [1164]
2190 IF a$<>"" THEN PRINT a$;                        [964]
2200 GOTO 2170                                       [361]
2210 '                                               [117]
2220 '>>> Grosstext <<<                              [1753]
2230 '                                               [117]
2240 PRINT"Achtung : Der Inhalt der oberst           [5880]
en Text- zeile wird geloescht !!!"
2250 PRINT:PRINT                                     [743]
2260 PRINT"Wollen Sie die Grosstext-Funkti           [5221]
on zu-   ruecknehmen (J/N) ?"
2270 a$=LOWER$(INKEY$)                               [960]
2280 IF a$="j" THEN 350                              [1092]
2290 IF a$<>"n" THEN 2270                            [727]
2300 CLS:LINE INPUT "Text : ";t$                     [1669]
2310 INPUT "Vergroesserung (2-10) ";v                [2731]
2320 IF v<2 OR v>10 THEN PRINT CHR$(7):GOT           [2175]
O 2310
2330 v=v*2                                           [726]
2340 |IN:xp=0:yp=0:f1=TEST(xp,yp)                    [1495]
2350 GOSUB 660                                       [881]
2360 PLOT xp,yp,f1                                   [1383]
2370 LOCATE 1,1:PRINT t$                             [716]
2380 x1=16*LEN(t$)                                   [1538]
2390 mx=xp:my=yp                                     [1427]
2400 FOR x=0 TO x1 STEP 2                            [1026]
2410   FOR y=384 TO 398 STEP 2                       [1350]
2420     f2=TEST(x,y)                                [863]
2430     mx1=mx+(x/2)*v                              [581]
2440     my1=my-((398-y)/2)*v                        [1392]
2450     FOR x1=mx1 TO mx1+v                         [1513]
2460       PLOT x1,my1:DRAWR 0,-v,f2                 [1752]
2470     NEXT x1                                     [475]
2480   NEXT y                                        [359]
```

Listing MICRO PAINT

```
2490 NEXT x                                          [356]
2500 LOCATE 1,1:PRINT SPACE$(LEN(t$))                [2193]
2510 f1=TEST(xp,yp):GOTO 2350                        [1675]
2520 '                                               [117]
2530 '>>> Laden <<<                                  [1153]
2540 '                                               [117]
2550 PRINT"L a d e n"                                [728]
2560 PRINT"---------"                                [1011]
2570 PRINT:PRINT                                     [743]
2580 PRINT"Directory anschauen (J/N) ?"              [3969]
2590 a$=LOWER$(INKEY$)                               [960]
2600 IF a$="n" THEN 2630                             [719]
2610 IF a$<>"j" THEN 2590                            [1552]
2620 CAT                                             [96]
2630 LINE INPUT "Name des Screens : ";na$            [2557]
2640 PRINT"Legen Sie die Kassette/Diskette           [7487]
 mit ";na$;"ein und druecken Sie eine Tast
e !"
2650 WHILE INKEY$<>"":WEND                           [1786]
2660 WHILE INKEY$="":WEND                            [1607]
2670 PRINT"Achtung : das Aktuelle Bild wir           [9460]
d durch das Einladen eines neuen Screens
ge-     loescht."
2680 PRINT"Wolle Sie das Bild wirklich lad           [4971]
en (J/N)?"
2690 a$=LOWER$(INKEY$)                               [960]
2700 IF a$="n" THEN 350                              [752]
2710 IF a$<>"j" THEN 2690                            [1348]
2720 LOAD na$,&C000                                  [633]
2730 |OUT                                            [408]
2740 GOTO 350                                        [470]
2750 '                                               [117]
2760 '>>> Speichern <<<                              [529]
2770 '                                               [117]
2780 PRINT"S p e i c h e r n"                        [982]
2790 PRINT"-----------------"                        [1295]
2800 PRINT:PRINT                                     [743]
2810 LINE INPUT "Name des Screens : ";a$             [2498]
2820 PRINT"Legen Sie die Kassette/Diskette           [7513]
, auf die";a$;"gespeichert werden soll"
2830 PRINT"und druecken Sie eine Taste !"            [3191]
2840 WHILE INKEY$<>"":WEND                           [1786]
2850 WHILE INKEY$="":WEND                            [1607]
2860 |IN                                             [361]
2870 SAVE a$,b,&C000,&4000                           [1116]
2880 GOTO 350                                        [470]
2890 '                                               [117]
2900 '>>> Symbole setzen <<<                         [1720]
2910 '                                               [117]
2920 DATA Joystick,Computer,Monitor,Tasche           [9814]
nrechner,Diskette,Buch,Schneider-Zeichen,C
PC-Zeichen,Diskettenlaufwerk 1,Diskettenla
ufwerk 2,Wolke 1,Wolke 2,Wolke 3,Stuhl 1,S
tuhl 2,Tisch,Haus
2930 CLS:RESTORE 2920                                [579]
2940 FOR x=1 TO 17                                   [820]
2950   READ a$:PRINT x;".) ";a$                      [1050]
2960 NEXT                                            [350]
2970 PRINT:PRINT STRING$(80,"=")                     [1793]
2980 PRINT                                           [361]
2990 INPUT "Welche Symbolnummer ";sym                [1799]
3000 IF sym<>INT(sym) OR sym<1 OR sym>17 T           [3585]
HEN PRINT CHR$(7):GOTO 2920
3010 CLS                                             [91]
3020 xp=320:yp=200                                   [1171]
3030 ON sym GOSUB 3180,3250,3310,3360,3460           [3680]
,3510,3560,3610,3740,3790,3820,3840,3910,3
960,3980,4020,4040
3040 FOR x=1 TO 2000:NEXT                            [670]
3050 CLS:|IN                                         [333]
3060 f1=TEST(xp,yp)                                  [584]
3070 GOSUB 660                                       [881]
3080 ON sym GOSUB 3180,3250,3310,3360,3460           [3680]
,3510,3560,3610,3740,3790,3820,3840,3910,3
960,3980,4020,4040
3090 a$=LOWER$(INKEY$)                               [960]
3100 IF a$="z" THEN |IN:GOTO 3070                    [1680]
3110 IF a$<>"s" THEN 3090                            [1461]
3120 |OUT                                            [408]
3130 f1=TEST(xp,yp)                                  [584]
3140 GOTO 3070                                       [459]
3150 '                                               [117]
3160 'Symbole zeichnen                               [2040]
3170 '                                               [117]
3180 PLOT xp,yp,f:DRAWR-4,4:PLOTR 0,2:DRAW           [5664]
R 4,4:DRAWR 30,0:DRAWR 4,-4:PLOTR 0,-2:DRA
WR-4,-4:DRAWR-30,0
3190 PLOT xp+4,yp:DRAWR-4,-4:DRAWR 10,0:DR           [3159]
AWR-4,4
3200 PLOT xp+26,yp:DRAWR 4,-4:DRAWR-10,0:D           [2968]
RAWR 4,4
```

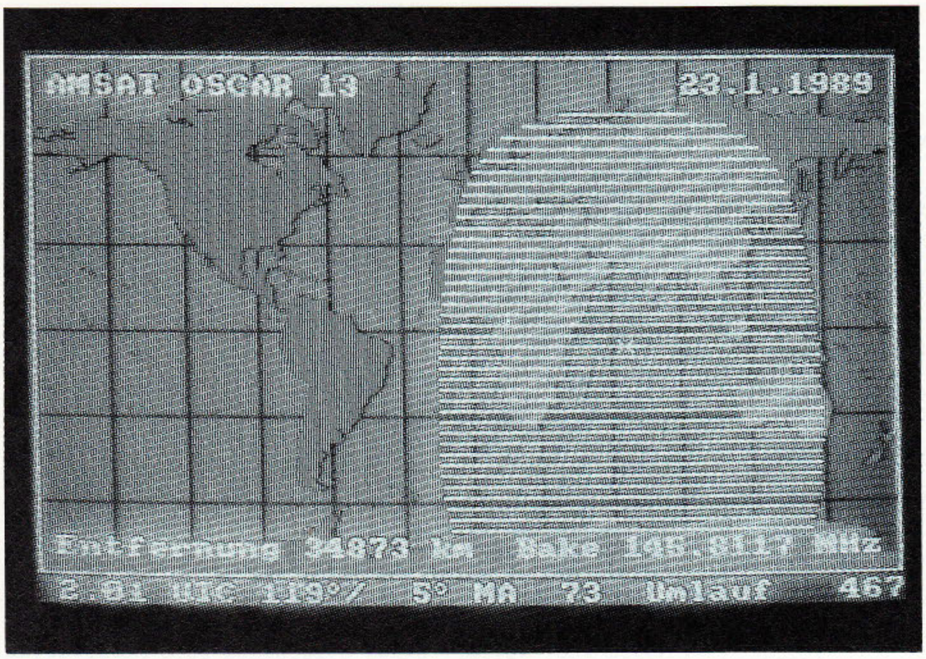
Listing MICRO PAINT

```
3210 PLOT xp+4,yp+12:DRAWR 6,0:PLOT xp+26,  [4225]
yp+12:DRAWR-6,0
3220 PLOT xp+4,yp+14:DRAWR 6,0:PLOT xp+26,  [2784]
yp+14:DRAWR-6,0
3230 PLOT xp+12,yp+12:DRAWR 0,20:DRAWR-4,4  [4546]
:DRAWR 4,4:DRAWR 6,0:DRAWR 4,-4:DRAWR-4,-4
:DRAWR 0,-20
3240 RETURN                                  [555]
3250 PLOT xp,yp,f:DRAWR 52,0:DRAWR 0,20:DR  [2904]
AWR-52,0:DRAWR 0,-20
3260 FOR x=4 TO 50 STEP 4                   [1182]
3270    PLOT xp+x,yp+4                       [592]
3280    PLOTR 0,4:PLOTR 0,4:PLOTR 0,4       [1308]
3290 NEXT x                                  [356]
3300 RETURN                                  [555]
3310 PLOT xp,yp,f:DRAWR 40,0:DRAWR 0,30:DR  [3258]
AWR-40,0:DRAWR 0,-30
3320 MOVER 4,8:DRAWR 4,-4:DRAWR 24,0:DRAWR  [4778]
 4,4:DRAWR 0,14:DRAWR-4,4:DRAWR-24,0:DRAWR
-4,-4:DRAWR 0,-14
3330 PLOT xp+10,yp+20:DRAWR 4,4:DRAWR 4,-4  [3961]
:DRAWR 4,4:DRAWR 4,-4:DRAWR 4,4
3340 PLOT xp+10,yp+12:DRAWR 4,4:DRAWR 4,-4  [3616]
:DRAWR 4,4:DRAWR 4,-4:DRAWR 4,4
3350 RETURN                                  [555]
3360 PLOT xp,yp,f:DRAWR 36,0:DRAWR 0,52:DR  [3434]
AWR-36,0:DRAWR 0,-52
3370 FOR x=4 TO 28 STEP 8                    [637]
3380    FOR y=4 TO 36 STEP 8                [1179]
3390      PLOT xp+x,yp+y                     [799]
3400      DRAWR 4,0:DRAWR 0,4                [776]
3410      DRAWR-4,0:DRAWR 0,-4              [1533]
3420    NEXT y                               [359]
3430 NEXT x                                  [356]
3440 MOVER 2,8:DRAWR-24,0:DRAWR 0,4:DRAWR   [2995]
24,0:DRAWR 0,-4
3450 RETURN                                  [555]
3460 PLOT xp,yp,f:DRAWR 20,0:DRAWR 0,30:DR  [3060]
AWR-20,0:DRAWR 0,-30
3470 MOVER 0,4:DRAWR 20,0:MOVER 0,4:DRAWR-  [5245]
20,0:MOVER 0,4:DRAWR 20,0:MOVER 0,2:DRAWR-
20,0
3480 PLOT xp+10,yp+18:DRAWR-2,2:DRAWR 0,4:  [5871]
DRAWR 2,2:DRAWR 2,-2:DRAWR 0,-4:DRAWR-2,-2
3490 PLOT xp+4,yp+26:PLOTR 12,0             [1507]
3500 RETURN                                  [555]
3510 PLOT xp+10,yp,f:DRAWR-10,10:DRAWR 0,2  [5431]
0:DRAWR 10,-10:DRAWR 10,10:DRAWR 0,-20:DRA
WR-10,-10:DRAWR 0,20
3520 MOVER 0,2:DRAWR-8,8:MOVER 16,0:DRAWR-  [2682]
8,-8
3530 PLOT xp+12,yp+20:DRAWR 0,-2:PLOTR 2,2  [2417]
:PLOTR 2,0:PLOTR 2,-2
3540 PLOT xp+12,yp+14:DRAWR 0,-2:PLOTR 2,0  [3580]
:PLOTR 2,4:PLOTR 2,-2
3550 RETURN                                  [555]
3560 FOR y=0 TO 1                           [1181]
3570 PLOT 2+xp+10,2*y+yp+20,f:DRAWR-14,0:D  [6829]
RAWR-2,-4:DRAWR-2,-2:DRAWR-2,0:DRAWR 2,0:D
RAWR 2,-2:DRAWR 2,-4:DRAWR 14,0
3580 PLOTR 6,0:DRAWR 14,0:DRAWR 2,-4:DRAWR  [6213]
 2,-2:DRAWR 2,0:DRAWR-2,0:DRAWR-2,-2:DRAWR
-2,-4:DRAWR-14,0
3590 NEXT                                    [350]
3600 RETURN                                  [555]
3610 FOR x=0 TO 180 STEP 5:PLOT 10+xp+20*S  [4370]
IN(x),yp+20*COS(x),f:NEXT
3620 DRAWR-40,0                              [636]
3630 FOR x=180 TO 360 STEP 5:PLOT (xp-30)+  [4266]
20*SIN(x),yp+20*COS(x),f:NEXT
3640 DRAWR 40,0                              [608]
3650 MOVE xp,yp+2:FILL f                     [858]
3660 PLOT xp-30,yp+16,0:DRAWR -16,-16:DRAW  [8598]
R 16,-16:PLOTR 2,0:DRAWR-16,16:DRAWR 16,16
:PLOTR 2,0:DRAWR-16,-16:DRAWR 16,-16
3670 PLOT xp-20,yp+16:DRAWR 0,-32:PLOTR 2,  [3298]
0:DRAWR 0,32
3680 FOR x=0 TO 2                            [712]
3690 DRAWR 8,0:DRAWR 4,-4:DRAWR 0,-4:DRAWR  [1589]
-4,-4:DRAWR-8,0
3700 MOVER 0,10                              [649]
3710 NEXT                                    [350]
3720 PLOT xp+10,yp+16:DRAWR -16,-16:DRAWR   [6645]
16,-16:PLOTR 2,0:DRAWR-16,16:DRAWR 16,16:P
LOTR 2,0:DRAWR-16,-16:DRAWR 16,-16
3730 RETURN                                  [555]
3740 PLOT xp,yp,f:DRAWR 16,0:DRAWR 0,16:DR  [3559]
AWR-16,0:DRAWR 0,-16
3750 MOVER 4,12:DRAWR 8,0                   [1127]
3760 PLOT xp+4,yp+8:PLOTR 4,-4:PLOTR 4,4:D  [3572]
RAWR 4,0
```

Listing MICRO PAINT

```
3770 PLOT xp+16,yp:DRAWR 20,20:DRAWR 0,16:  [4498]
DRAWR-20,-20:MOVER-16,0:DRAWR 20,20:DRAWR
16,0
3780 RETURN                                  [555]
3790 PLOT xp,yp,f:DRAWR 26,0:DRAWR 0,26:DR  [4104]
AWR-26,0:DRAWR 0,-26
3800 PLOT xp+4,yp+22:DRAWR 18,0:PLOTR 0,-2  [5784]
:DRAWR-18,0:PLOTR 0,-4:PLOTR 6,-6:PLOTR 6,
4:DRAWR 6,0
3810 RETURN                                  [555]
3820 PLOT xp,yp,f:DRAWR 8,4:DRAWR 4,-4:DRA [10046]
WR 6,8:DRAWR 8,-4:DRAWR 6,16:DRAWR-10,-4:D
RAWR-4,10:DRAWR-14,0:DRAWR 0,-12:DRAWR-8,0
:DRAWR 4,-8:DRAWR 0,-4
3830 RETURN                                  [555]
3840 GOSUB 3820                             [1005]
3850 MOVE xp+10,yp+8:FILL 1                 [1263]
3860 PLOT xp+30,yp+18,f                     [1082]
3870 FOR x=xp TO xp+24 STEP 6               [1435]
3880 PLOT x,yp:DRAWR-14,-14                 [1526]
3890 NEXT                                    [350]
3900 RETURN                                  [555]
3910 GOSUB 3820                             [1005]
3920 MOVE xp+10,yp+8:FILL 1                 [1263]
3930 PLOT xp+30,yp+18,f                     [1082]
3940 PLOT xp+14,yp:DRAWR-24,-40:DRAWR 24,2  [3958]
4:DRAWR-24,-40:DRAWR-8,16:MOVER 24,-8:DRAW
R-16,-8
3950 RETURN                                  [555]
3960 PLOT xp,yp,f:DRAWR 0,20:DRAWR 20,0:MO  [2918]
VER 0,-20:DRAWR 0,40
3970 RETURN                                  [555]
3980 PLOT xp+8,yp,f:DRAWR 0,20:DRAWR 32,0:  [5003]
MOVER 0,-20:DRAWR 0,20:DRAWR-4,20:DRAWR-24
,0:DRAWR-4,-20
3990 MOVER 4,20:DRAWR 0,20:MOVER 24,0:DRAW  [3690]
R 0,-20:MOVER 0,8:DRAWR-24,0:MOVER 0,8:DRA
WR 24,0
4000 MOVE xp+20,yp+32:FILL f                [1627]
4010 RETURN                                  [555]
4020 PLOT xp,yp,f:DRAWR 0,10:DRAWR 30,0:DR  [7495]
AWR 0,-10:MOVER 0,10:DRAWR 10,20:DRAWR-30,
0:DRAWR-10,-20:MOVER 40,20:DRAWR 0,-16
4030 RETURN                                  [555]
4040 PLOT xp,yp,f:DRAWR 10,0:DRAWR 0,20:DR  [5655]
AWR-10,0:DRAWR 0,-20:MOVER 0,20:DRAWR 4,10
:PLOTR 2,0:DRAWR 4,-10
4050 DRAWR 20,10:DRAWR-4,10:DRAWR-20,-10:M  [2567]
OVE xp+10,yp:DRAWR 20,10:DRAWR 0,20
4060 RETURN                                  [555]
4070 '                                       [117]
4080 '>>> Drucken <<<                       [1007]
4090 '                                       [117]
4100 PRINT"MM-COPY geladen (J/N) ?"         [2444]
4110 a$=LOWER$(INKEY$)                       [960]
4120 IF a$="n" THEN 350                      [752]
4130 IF a$<>"j" THEN 4110                   [1632]
4140 CLS                                      [91]
4150 PRINT TAB(4);"1.) Mini-Format  (7.0 *  [3070]
5.0 cm)"
4160 PRINT TAB(4);"2.) Micro-Format (3.5 *  [1951]
2.5 cm)
4170 PRINT:PRINT                             [743]
4180 PRINT STRING$(80,"*")                  [1446]
4190 PRINT                                   [361]
4200 PRINT"Bitte waehlen (1/2) !"           [3069]
4210 a$=INKEY$                               [278]
4220 IF a$="1" THEN au$="ni":GOTO 4250      [1074]
4230 IF a$<>"2" THEN 4210                    [867]
4240 au$="cro"                               [779]
4250 PRINT:PRINT                             [743]
4260 PRINT"Bild im mi";au$;"-Format ausdru  [4159]
cken."
4270 PRINT"Sind Sie sicher (J/N) ?"         [1463]
4280 a$=LOWER$(INKEY$)                       [960]
4290 IF a$="n" THEN 350                      [752]
4300 IF a$<>"j" THEN 4280                   [1702]
4310 MODE 1:|IN                              [597]
4320 IF au$="ni" THEN |MINI.COPY ELSE |MIC  [2362]
RO.COPY
4330 GOTO 350                                [470]
4340 '                                       [117]
4350 '>>> Loeschen <<<                       [634]
4360 '                                       [117]
4370 CLS:PRINT"Loeschen - sind Sie sicher   [2697]
(J/N) ?"
4380 a$=LOWER$(INKEY$)                       [960]
4390 IF a$="n" THEN 350                      [752]
4400 IF a$<>"j" THEN 4380                   [1698]
4410 |CLS:GOTO 350                          [1520]
```

Listing MICRO PAINT



Blick zu den Sternen

SAT-MASTER – Satellit-Berechnungsprogramm

Leider ist die Anwender-Software für den CPC im Bereich des Amateurfunks sehr spärlich gesät. Eigentlich schade, denn gerade der CPC (6128) fällt durch gute Abschirmung gegen Hochfrequenz, im Vergleich zu anderen Rechnern seiner Preisklasse, auf, was für mich das Hauptkriterium zum Kauf dieses Computers darstellte, da ich Funkamateur bin und den Computer als Hilfsmittel zum Hobby verwenden wollte, ohne beim Kauf arm zu werden. Wie sich später herausstellte, war aber keine Software erhältlich, was mich wiederum zur Selbsthilfe veranlaßte.

Hier liegt nun das Ergebnis dieser Selbsthilfe vor. 'SAT' ist ein Programm zum Berechnen von Satellitenbahnen, der Sendebereich wird dabei grafisch ausgegeben.

Wie das Programm funktioniert

Zum Installieren des Programms starten Sie zunächst das Hilfsprogramm "SAT.GEN". Es generiert nun die Datei "SAT.DAT" auf der Diskette. Diese Datei enthält einige Daten, die das Hauptprogramm benötigt. In ihr stehen die eingegebene eigene Position sowie die Bahndaten der Satelliten. Als nächstes wird die Weltkarte mit "SAT.MAP" erzeugt. Dieses Programm enthält alle Daten für die Karte, die das Hauptprogramm zur Darstellung der Hörbarkeitszone der Satelliten verwendet. Die Karte wird erzeugt, indem die Kontinente umrandet und dann mit Hilfe der Fill-Routine ausgefüllt werden. Die Daten setzen sich aus dem ersten Punkt, der Anzahl der weiteren Punkte, der Anzahl der Punkte zum Füllen, den Umrandungspunkten und den Füllpunkten zusammen. Anschließend folgen die Inseln. Alle Punkte werden in Grad Ost und Grad Nord angegeben. Wem die Karte also nicht gefällt, der kann sie gerne ändern, aber Vorsicht: eine falsche Angabe kann fatale Folgen haben!

Wenn die Karte fertig gemalt ist, kann sie durch die Eingabe von "s" als "SAT.SCR" gespeichert werden. Jede andere Eingabe beendet das Programm ohne Sicherung der Karte. Danach wird "SAT." eingegeben. In Zeile 20 wird der Monitor auf 60 Hz umgeschaltet. Sollte das bei Ihnen zu Komplikationen führen, so löschen Sie einfach diese Zeile. Sie ist für das weitere Programm bedeutungslos. 'SAT' ist der Starter, der die Karte und Bahndaten lädt, den Blockverschiebebefehl COPY – entspricht dem Assemblerbefehl LDIR – installiert und die Sonderzeichen generiert. Er startet anschließend das eigentliche Hauptprogramm "SAT.PRG". Das Programm wird also mit "RUN"SAT" gestartet. Es meldet sich nun mit dem Hauptmenü.

Die Dateneingabe

Bei der Bahnberechnung fragt das Programm zunächst nach dem Satelliten und dann nach dem Datum, getrennt nach Tag, Monat und Jahr, sowie nach der Uhrzeit in UTC. Mit UTC ist die Weltzeit gemeint, früher bekannt als GMT. Das ist unsere Zeit -1 Stunde beziehungsweise Sommerzeit -2 Stunden. Damit hat das Programm alle notwendigen Daten und berechnet nun die Position des Satelliten. Links oben erscheint der Name des Satelliten, rechts oben das Datum und unten die Statuszeile. Sie enthält die Uhrzeit, die Antennenwinkel Azimut und Elevation, die momentane Anomalie MA und die Umlaufnummer.

Die Uhrzeit wird in UTC, die Antennenwinkel in Grad angegeben. Azimut gibt die Richtung von 0 bis 360 Grad von Nord ausgehend im Uhrzeigersinn, Elevation den Erhebungswinkel über Horizont von -90 bis 90 Grad an. Die Angabe 180 Grad / 10 Grad bedeutet also, daß der Satellit genau im Süden 10 Grad über dem Horizont steht. Die momentane Anomalie MA gibt an, auf welchem Punkt der Bahn, ausgehend vom Perigäum (dem erdnächsten Punkt), sich der Satellit gerade befindet. Sie wird in 256stel Teilen angegeben. Eine MA von 0 bedeutet, daß der Satellit sehr dicht an der Erde ist, eine MA von 128, daß er sich im Apogäum (erdfernsten Punkt) befindet. Diese Angabe ist wichtig, da sie die Lage des Satelliten im All indirekt angibt, was wichtig für die Stromversorgung des Satelliten ist, die mittels Solarzellen geschieht. Liegt der Satellit nämlich ungünstig zur Sonne, so kann er zeitweilig von der Bodenstation abgeschaltet werden, um die Akkus zu schonen.

Ist die Elevation größer als -6 Grad, so erscheinen unten in der Karte Angaben über die ungefähre Entfernung des Satelliten vom eigenen Standort sowie die ungefähre Bakenfrequenz. Bei der ersten Angabe erscheint die genaue Sendefrequenz des Satelliten, alle weiteren Angaben berücksichtigen den Dopplereffekt. Fehlt die Bakenfrequenz in der Anzeige, so wurde sie bei der Bahndateneingabe nicht mit angegeben. Der schraffierte Bereich auf der Karte ist der Sichtbereich des Satelliten. Das Programm berechnet ungefähr alle 30 Sekunden die neuesten Werte. Sollte die Uhr falsch gehen oder soll einige Minuten im voraus berechnet

werden, so ist ein Verstellen der Uhrzeit mit den Cursortasten in Schritten zu einer beziehungsweise zehn Minuten möglich.

Neue Berechnung

Durch "0" gelangt man in das Hauptmenü zurück. Wird erneut eine Berechnung gewünscht, so kann die alte Zeit durch <ENTER> bei der Frage nach dem Tag übernommen werden. Mit dem Menüpunkt "Satellit einfügen" kann ein neuer Satellit in die Datentabelle übernommen werden. Das Programm fragt zunächst nach dem Namen des Satelliten. Eine Eingabe von "0" bricht diesen Programmteil hier ab, sonst wird anschließend nach den Kepler-Bahndaten gefragt, wobei die Eingabe der 'Epoch Time' getrennt nach Jahr und Tag erfolgt. Die ersten beiden Ziffern dieses Wertes geben das Jahr, alle weiteren den Tag an. Liegt Ihnen direkt die Referenzzeit vor, so können Sie diese auch direkt eingeben. Beantworten Sie dazu die Frage nach den Epoche Jahr mit <ENTER>. Mit dem Menüpunkt "Satellit löschen" können Sie einen Satelliten aus der Datei entfernen. Er wird vollständig gelöscht. Ein Retten ist nur durch Herausnehmen der Diskette, Reset und Neustart möglich, wenn die Daten bereits auf Diskette stehen, da die jeweils aktuellen Daten beim Programmende automatisch auf Diskette gespeichert werden.

Mit dem Menüpunkt "Bahndaten ändern" haben Sie die Möglichkeit, alte oder falsche Bahndaten zu ändern. Die alten Werte können durch Eingabe von <ENTER> übernommen werden, neue Werte müssen vollständig eingegeben werden. Sollte bei den Bahndaten EMM beziehungsweise die Bakenfrequenz fehlen, so können Sie hier auch 0 eingeben. Übrigens: Bahndaten werden im Laufe der Zeit ungenau, da sich die Fehler summieren. Sie sollten sich alle paar Monate neue Bahndaten gönnen, wenn die Berechnung hinreichend genau bleiben soll.

Mit dem Menüpunkt "Satellit umnennen" können Sie den Namen eines Satelliten ändern, wobei die Bahndaten erhalten bleiben. Eine Eingabe von "0" bricht auch diesen Menüpunkt ab.

Mit dem Menüpunkt "Standort ändern" können Sie einen Bezugspunkt auf der Erde zur Berechnung der Antennenrichtung und Entfernung eingeben. Diese Position wird mit den Bahndaten auf der Diskette gespeichert, das heißt, es genügt, einmal den eigenen Wohnort anzugeben. Wenn Sie das Programm beenden und Sie einige Daten geändert haben, so achten Sie darauf, daß sich die Diskette im Laufwerk befindet, da das Programm versuchen wird, die neuen Daten zu speichern.

Probleme mit dem CPC 464

Per Emulator ist das Programm zwar theoretisch auf allen CPC 464 lauffähig, doch könnten durch die unterschiedlichen Versionen dieses Computers (Baujahr vor oder nach 1985) Probleme auftauchen. In diesem Fall ist Handarbeit angesagt.

So, damit wäre alles Wichtige genannt, wir wünschen Ihnen viel Vergnügen beim Beobachten 'Ihres' Satelliten.

(H.-H. Schulze / jb)

für 464-664-6128

```
10 SYMBOL AFTER 90:MEMORY &55FF:FOR i=&AFO [12593]
0 TO &AF2D:READ a$:POKE i,VAL("&"+a$):NEXT
 i:CALL &AF00:DEG:DATA 1,9,AF,21,12,AF,C3,
D1,BC,E,AF,C3,16,AF,43,4F,50,D9,0,0,0,0,FE
,3,C0,DD,5E,0,DD,56,1,DD,4E,2,DD,46,3,DD,6
E,4,DD,66,5,ED,B0,C9
20 FOR i=&B0C7 TO &B0EF:READ a$:POKE i,VAL [11221]
("&"+a$):NEXT:CALL &B0C7:DATA F5,C5,E5,21,
F0,B0,1,F,BC,ED,49,2B,7E,4,ED,79,5,D,F2,D0
,B0,E1,C1,F1,C9,3F,28,2E,8E,1F,6,19,1B,0,7
,0,0,30,0,C0,0
30 KEY DEF 66,0,0,0,0:SYMBOL 123,198,0,120 [10775]
,12,124,204,118,0:SYMBOL 125,198,0,102,102
,102,102,62,0:SYMBOL 124,198,0,60,102,102,
102,60,0:SYMBOL 126,120,198,198,252,198,19
8,248,192:SYMBOL 91,219,60,102,102,126,102
,102,0
40 SYMBOL 255,48,72,72,48,0,0,0,0:SYMBOL 9 [11006]
3,198,0,198,198,198,198,60,0:SYMBOL 92,198
,56,198,198,198,108,56,0:SYMBOL 163,24,36,
36,24,0,0,0,0:KEY DEF 17,1,123,91:KEY DEF
19,1,125,93:KEY DEF 26,1,124,92:KEY DEF 24
,1,126:KEY DEF 24,1,94,163
50 CALL &BB48:POKE &BDEE,&C9:MODE 1:BORDER [15729]
 0:PAPER 0:ORIGIN 320,200:p0=1:p1=9:p2=3:p
3=26:INK 0,0:INK 1,0:INK 2,0:INK 3,0:a$="s
at.scr":LOAD a$:PLOT -317,-180,3:DRAW 317,
-180,3:PLOT -319,-182,0:DRAW 319,-182,0:PL
OT -319,-184,0:DRAW 319,-184,0
60 LOCATE 1,25:PRINT STRING$(40," ");:INK  [15576]
0,p0:INK 1,p1:INK 2,p2:INK 3,p3:¦COPY,&C00
0,&4000,&5600:ORIGIN 0,0:PLOT 85,320,3:DRA
W 85,220:PLOT 85,270:DRAW 125,270:DRAW 125
,320:DRAW 125,220:PLOT 155,220:DRAW 175,32
0:DRAW 195,220:PLOT 165,270
70 DRAW 185,270:PLOT 225,220:DRAW 225,320: [12572]
DRAW 245,270:DRAW 265,320:DRAW 265,220:PLO
T 295,310:DRAW 295,280:DRAW 305,270:DRAW 3
25,270:DRAW 335,260:DRAW 335,230:DRAW 325,
220:DRAW 305,220:DRAW 295,230:PLOT 295,310
:DRAW 305,320:DRAW 325,320
80 DRAW 335,310:PLOT 365,310:DRAW 375,320: [12439]
DRAW 395,320:DRAW 405,310:DRAW 405,230:DRA
W 395,220:DRAW 375,220:DRAW 365,230:DRAW 3
65,310:PLOT 435,320:DRAW 435,220:PLOT 475,
320:PLOT 475,320:DRAW 435,320:PLOT 435,270
:DRAW 465,270:PLOT 495,320
90 DRAW 535,320:PLOT 515,320:DRAW 515,220: [12992]
PLOT 535,330:DRAW 545,320:DRAW 545,220:DRA
W 535,210:DRAW 85,210:DRAW 75,220:DRAW 75,
320:DRAW 85,330:DRAW 535,330:PLOT 535,340:
DRAW 555,320:DRAW 555,220:DRAW 535,200:DRA
W 85,200:DRAW 65,220
100 DRAW 65,320:DRAW 85,340:DRAW 535,340:P [6970]
EN 3,1:LOCATE 15,14:PRINT "pr{sentiert":LO
CATE 11,18:PRINT "S A T - M A S T E R":PLO
T 157,108:DRAW 465,108
110 maxsat=20:DIM sat$(maxsat),zi(maxsat), [7603]
zo(maxsat),ze(maxsat),zw(maxsat),zmm(maxsa
t),zemm(maxsat),zg(maxsat),zumlauf(maxsat)
,zbake(maxsat)
120 PEN 3,0:ON BREAK STOP:f=360.985647:max [15303]
sat$="123456789ABCDEFGHIJK":OPENIN "sat.da
t":INPUT #9,sat,meml,memb:FOR i=1 TO sat:I
NPUT #9,sat$(i),zi(i),zo(i),ze(i),zw(i),zm
m(i),zemm(i),zg(i),zumlauf(i),zbake(i):NEX
T i:CLOSEIN:CHAIN"SAT.PRG"
```

Listing SAT

```
10 'Dieser Lader generiert "SAT.DAT" mit d [8294]
em Standort 9 Grad Ost und 52 Grad Nord, s
owie den beiden Satelliten Oscar 10 und Os
car 13
20 OPENOUT "sat.dat":WRITE #9,2,9,52:'Anza [5443]
hl der Satelliten,Laenge,Breite
30 WRITE #9,"AMSAT OSCAR 10       ",27.3293 [5843]
,323.874,0.6026471,304.2954,2.05875576,0.0
000008,3786.83715,3701,145.81:'Oscar 10-Ba
hndaten
```

Listing SAT

```
40 WRITE #9,"AMSAT OSCAR 13      ",57.4,24 [6788]
8.11,0.6536,186.95,2.09695,0,3839.65788,47
,145.812:'Oscar 13-Bahndaten vom 13.7.1988
50 CLOSEOUT                               [902]
60 'Es handelt sich um Amateurfunk-Satelli [6997]
ten, somit besteht eine allgemeine Empfang
sgenehmigung!
```

```
10 CLS                                    [91]
20 ' *** Weltkarte ***                   [1904]
30 ' *** Afrika ***                      [1714]
40 DATA -6,36,149,1:' Init                [720]
50 DATA -7,34,-8,33.5,-9,33,-10,30,-11,29, [4225]
-12,28,-13,28,-14,27,-15,25.5,-16,24.3
60 DATA -17,22,-17.5,21.5,-17,20.5,-16.5,1 [4006]
8,-17,16,-18,15,-17,13,-17,12,-16,12,-15,1
1
70 DATA -14,10,-13,8.5,-12,7,-11,6,-10,5.5 [4071]
,-9,5,-8,4.5,-7,4,-6,5,-5,4.5
80 DATA -4,4.5,-3,5,-2,4.5,-1,5,0,5.5,1,6, [3681]
2,6.5,4,6.5,5,5,6,4.5
90 DATA 8,4.5,9,4,10,2,9,-0.5,9.5,-2,10,-8 [1966]
,11,-2.5,12,-5,13,-8,13.5,-9
100 DATA 14,-9.5,13.5,-10,13.5,-11.5,13,-1 [4616]
2.5,12,-15.5,11.5,-16,11.5,-18,12,-18.5,13
,-20,13.5,-21.5
110 DATA 14,-22.5,14.5,-25,14.5,-26,15,-28 [3858]
,15.5,-28,16,-29,17,-29.5,17,-30,18,-31.5,
18,-32
120 DATA 18,-33,16.5,-33,18,-34,20,-35,22, [2665]
-34,29,-34,25,-33.5,27,-33.5,28,-33,29,-32
130 DATA 30,-31.5,31,-29.5,32,-28.5,32.5,- [4230]
28,32.5,-26,33,-25.5,35,-25,36,-24,36,-22,
35,-20.5
140 DATA 36,-19,37,-18,39,-17,40,-16,40.5, [4046]
-15.5,40.5,-10,39.5,-10,39.5,-8,39.5,-4,40
,-3.5
150 DATA 41,-2.5,42,-1,43,0,45,2,46,2.5,47 [2240]
,3,48,5,49,6,50,9,51,10
160 DATA 51,12,50,11.5,48,11.5,47,11,46,11 [2815]
,45,10.5,44,11,43,12,43,13,42,14
170 DATA 40,16,39,17,38,18.5,37.5,19,37.5, [2744]
21,36,22.5,36,22.5,35,25,33,28.5,32,31
180 DATA 31,31.5,30,31,29,31,28,31.5,26,31 [2510]
.5,24.5,32.5,22,33,20,32,20,31,19,30.5
190 DATA 18,31,15,31,10,33,11,34,11,36,4,3 [2862]
6,-2,35,-4,35,-6,36
200 DATA 0,20:'fill                       [618]
210 ' *** Australien ***                 [1460]
220 DATA 142,-11,43,1:'init              [1231]
230 DATA 141,-12,141,-16,140,-18,136,-15,1 [3080]
37,-12,134,-12,133,-11,130,-11,130,-14,129
,-15
240 DATA 128,-15,127,-14,125,-14,122,-17,1 [4443]
22,-19,113,-22,113,-27,115,-32,116,-34,118
,-35
250 DATA 120,-34,123,-34,126,-32,131,-31,1 [4080]
34,-33,135,-35,138,-33,137,-36,138,-35,140
,-38
260 DATA 143,-39,144,-38,145,-39,150,-37,1 [4830]
53,-28,153,-26,148,-20,147,-18,146,-15,144
,-15
270 DATA 144,-13,143,-11,142,-11         [1564]
280 DATA 130,-25:'fill                    [390]
290 ' *** Suedamerika ***                 [843]
300 DATA -80,9,65,1:'init                [1514]
310 DATA -78,9,-77,8,-75,12,-72,13,-73,12, [3318]
-72,11,-70,11,-70,12,-70,11,-69,11
320 DATA -69,10,-62,10,-62,8,-57,6,-53,6,- [3692]
52,4,-50,2,-50,0,-45,-2,-45,-3
330 DATA -40,-3,-35,-5,-35,-8,-38,-13,-39, [3685]
-19,-42,-22,-44,-22,-48,-26,-48,-28,-54,-3
5
340 DATA -58,-34,-57,-36,-57,-37,-56,-37,- [3714]
57,-38,-62,-38,-62,-41,-65,-41,-65,-43,-64
,-43
350 DATA -65,-43,-65,-45,-68,-46,-66,-48,- [2620]
68,-51,-67,-53,-64,-54,-67,-55,-74,-53,-75
,-50
360 DATA -74,-37,-70,-18,-76,-15,-80,-7,-8 [3317]
1,-6,-81,-4,-80,-3,-81,-2,-80,1,-79,2
```

Listing SAT

```
370 DATA -79,3,-78,3,-77,6,-79,9,-80,9      [832]
380 DATA -60,-20:'fill                      [495]
390 ' *** Nordamerika und Groenland ***    [1353]
400 DATA -79,9,196,2:' Init                 [857]
410 DATA -81,8,-81,7,-86,10,-86,11,-88,13, [2715]
-92,14,-95,16,-96,15,-104,18,-106,20
420 DATA -105,21,-109,25,-112,31,-113,32,- [2959]
110,23,-112,24,-112,26,-115,28,-114,28,-11
5,30
430 DATA -118,34,-121,35,-124,39,-125,41,- [3876]
125,48,-128,51,-127,51,-131,55,-133,55,-13
6,58
440 DATA -148,61,-148,60,-152,59,-151,61,- [3733]
154,59,-153,58,-160,55,-165,54,-160,56,-15
8,58
450 DATA -157,59,-162,59,-162,60,-167,60,- [4336]
165,61,-166,62,-165,63,-161,64,-161,65,-16
7,65
460 DATA -168,66,-165,67,-162,66,-167,68,- [5076]
157,72,-136,69,-129,70,-114,68,-119,72,-12
3,71
470 DATA -125,72,-125,74,-121,75,-111,73,- [4562]
100,74,-100,75,-112,75,-124,76,-110,78,-10
8,77
480 DATA -108,79,-94,78,-96,81,-70,83,-55, [3815]
82,-35,84,-23,83,-15,83,-20,78,-18,75
490 DATA -22,73,-22,70,-27,68,-32,67,-35,6 [4114]
6,-40,65,-44,60,-45,60,-46,61,-48,61
500 DATA -52,64,-53,67,-51,69,-55,68,-55,7 [3632]
1,-51,71,-53,72,-55,72,-58,76,-69,76
510 DATA -72,78,-65,80,-73,79,-80,74,-67,7 [3167]
0,-66,68,-62,66,-64,65,-67,66,-66,65
520 DATA -65,64,-65,63,-67,62,-73,64,-78,6 [2857]
4,-78,65,-74,65,-72,67,-76,67,-76,70
530 DATA -82,70,-82,66,-85,65,-81,63,-87,6 [2688]
3,-87,66,-89,64,-94,62,-95,59,-94,59
540 DATA -94,57,-90,57,-85,55,-82,55,-82,5 [2344]
3,-80,51,-79,52,-79,55,-76,55,-79,59
550 DATA -78,60,-78,63,-74,63,-70,61,-70,5 [4056]
9,-66,59,-65,60,-60,55,-56,53,-55,52
560 DATA -56,50,-54,49,-53,47,-59,48,-58,5 [3238]
1,-60,50,-67,50,-70,47,-64,49,-64,46
570 DATA -60,45,-65,43,-65,44,-65,45,-70,4 [3167]
3,-70,42,-73,41,-76,37,-75,35,-81,32
580 DATA -82,31,-80,26,-81,25,-82,25,-83,2 [2701]
8,-83,29,-84,30,-85,29,-86,30,-89,30
590 DATA -89,28,-94,30,-97,27,-97,25,-98,2 [3548]
2,-96,19,-95,18,-91,19,-91,21,-87,22
600 DATA -89,15,-85,16,-84,15,-84,11,-82,8 [1632]
,-79,9
610 DATA -90,15,-82,27:' Fill               [643]
620 ' *** Kuba ***                          [763]
630 DATA -84,22,3,0:' Init                  [828]
640 DATA -82,23,-75,20,-77,20              [1068]
650 ' *** Hispaniola ***                    [492]
660 DATA -72,19,1,0,-70,19                  [695]
670 ' *** Europa und Asien ***             [2099]
680 DATA 32,32,271,6:' Init                [1491]
690 DATA 34,32,35,32,36,35,36,37,27,37,26, [2821]
41,29,41,31,41,33,42,35,42
700 DATA 40,41,42,42,46,45,48,47,54,48,55, [3246]
45,52,45,51,44,53,42,54,37
710 DATA 50,37,48,39,50,40,46,45,33,45,38, [3469]
45,39,47,35,46,36,45,33,45
720 DATA 34,46,31,46,29,45,28,43,29,42,26, [2788]
41,23,41,24,38,23,37,22,37
730 DATA 20,42,14,46,12,44,15,42,16,42,15, [2426]
42,18,40,17,41,16,40,17,39
740 DATA 16,38,15,38,15,37,13,38,16,38,16, [2920]
40,14,41,13,41,10,43,10,44
750 DATA 9,45,6,43,3,43,3,42,0,40,0,39,-2, [2051]
38,-5,38,-6,37,-7,38
760 DATA -9,37,-9,39,-10,39,-9,41,-9,43,-8 [2991]
,44,-1,43,-1,46,-2,47,-5,48
770 DATA -5,49,-2,49,-2,50,-2,49,2,50,2,51 [2707]
,4,52,5,53,7,54,9,54
780 DATA 9,56,10,57,10,56,13,56,12,53,21,5 [2286]
4,21,58,24,57,25,59
790 DATA 30,60,28,61,22,60,21,61,21,63,26, [3140]
66,22,66,20,64,17,62,17,60
800 DATA 17,60,17,59,16,56,14,56,14,56,12, [2071]
59,8,58,6,59,5,62
810 DATA 9,64,16,69,14,68,25,71,28,71,41,6 [2570]
8,41,67,39,66,33,67,35,64
```

Listing SAT

```
820 DATA 37,62,37,64,41,64,43,66,45,66,45, [3832]
68,45,66,61,68,55,72,68,76,68,77,55,75,52,
73
830 DATA 55,72,68,67,70,73,78,72,104,77,89 [3722]
,82,113,77,113,75,105,73,115,74
840 DATA 125,73,130,71,141,73,168,69,168,7 [3531]
0,174,70,180,68,180,65,177,65,179,63
850 DATA 170,60,163,60,161,55,157,51,157,5 [3621]
7,163,62,158,61,157,59,142,59,135,55
860 DATA 140,54,140,48,135,43,130,42,127,3 [3117]
9,128,35,126,34,126,38,125,38,125,40
870 DATA 122,39,123,41,116,39,118,37,122,3 [4182]
7,119,35,121,31,120,30,121,31,118,25
880 DATA 115,23,110,22,110,18,111,19,110,2 [3426]
0,110,22,108,22,106,19,109,15,109,12
890 DATA 105,8,105,11,100,14,99,9,103,6,10 [2162]
4,2,101,4,100,7,98,8,98,17
900 DATA 95,16,94,16,94,20,92,22,87,22,87, [3001]
21,86,20,85,20,83,17,83,16
910 DATA 82,16,80,15,80,13,79,12,79,11,78, [3028]
11,78,10,77,8,76,8,73,19
920 DATA 73,22,72,21,70,21,69,22,70,23,69, [3070]
23,66,25,61,25,57,26,57,27
930 DATA 56,27,55,26,52,27,51,28,50,30,48, [2841]
30,49,28,50,27,51,24,51,26
940 DATA 51,24,54,24,56,26,57,25,59,24,60, [2629]
23,58,19,55,17,52,16,52,15
950 DATA 44,13,43,15,43,17,39,21,39,23,38, [3297]
24,35,28,34,27.5,32,30,32,32
960 DATA 60,45,12,43,13,42,15,41,54,73,159 [1928]
,55:' Fill
970 '*** Asien/Westen ***                   [1818]
980 DATA -180,66,7,1:' Init                 [1666]
990 DATA -180,68,-174,67,-172,67,-171,66,- [2668]
173,65,-173,64,-180,66
1000 DATA -179,67:' Fill                    [707]
1010 ' *** Neuseeland-Nord ***              [1432]
1020 DATA 173,-38,9,1:' Init                [1199]
1030 DATA 174,-38,173,-39,174,-40,174,-41, [3850]
173,-41,174,-41,178,-37,175,-38,173,-34
1040 DATA 175,-39:' Fill                    [1576]
1050 ' *** Neuseeland-Sued ***              [1135]
1060 DATA 173,-43,5,2:' Init                [701]
1070 DATA 170,-44,165,-48,167,-48,171,-47, [2604]
173,-43
1080 DATA 170,-46,172,-44:' Fill            [1588]
1090 ' *** Island ***                       [551]
1100 DATA -24,65,8,1:' Init                 [1658]
1110 DATA -23,66,-21,65,-14,66,-13,65,-14, [3121]
64,-18,63,-23,64,-23,66
1120 DATA -18,64:' Fill                     [1576]
1130 ' *** Neuguinea ***                    [901]
1140 DATA 140,-8,23,5:' Init                [817]
1150 DATA 141,-9,143,-9,144,-8,146,-8,147, [2797]
-10,151,-11,147,-7,148,-7,148,-6,146,-6
1160 DATA 146,-5,144,-4,137,-2,135,-4,134, [3602]
-1,131,-1,133,-3,133,-5,132,-5,137,-5
1170 DATA 139,-7,138,-8,140,-8              [1332]
1180 DATA 145,-6,134,-4,133,-2,149,-10,147 [1656]
,-8:' Fill
1190 ' *** Antarktis ***                    [1192]
1200 DATA -180,-78,37,1:'Init               [1674]
1210 DATA -158,-77,-145,-77,-145,-76,-125, [4516]
-73,-100,-74,-102,-73,-72,-73,-74,-69,-70,
-68,-68,-69
1220 DATA -65,-65,-58,-63,-62,-68,-60,-75, [3357]
-44,-78,-33,-77,-11,-72,-3,-70,33,-70,33,-
69
1230 DATA 39,-70,40,-68,52,-66,70,-67,70,- [3154]
68,75,-70,83,-67,112,-66,140,-66,166,-70

1240 DATA 170,-72,162,-75,163,-78,180,-78, [3125]
180,-90,-180,-90,-180,-78
1250 DATA 0,-78:'fill                       [859]
1260 ' *** Madagaskar ***                   [2089]
1270 DATA 49,-12,11,1:' Init                [1040]
1280 DATA 50,-13,50,-15,47,-25,45,-26,44,- [3637]
25,43,-22,44,-20,43,-18,44,-17,47,-15,49,-
12
1290 DATA 46,-20:' Fill                     [426]
1300 ' *** Taiwan ***                       [439]
1310 DATA 121,22,1,0,122,24                 [1266]
1320 ' *** Java ***                         [527]
1330 DATA 105,-7,1,0,121,-8                 [1086]
```

Listing SAT

```
1340 ' *** Palawan ***                      [757]
1350 DATA 117,8,1,0,119,11                  [850]
1360 ' *** Luzon ***                        [1659]
1370 DATA 121,13,9,1:' Init                 [1185]
1380 DATA 121,18,122,18,122,16,121,16,121, [2762]
14,125,14,125,8,124,8,121,13
1390 DATA 123,11:' Fill                     [561]
1400 ' *** Borneo ***                       [966]
1410 DATA 119,6,8,1,117,7,112,2,109,2,110, [3136]
-3,116,-4,118,1,117,3,119,6,112,0
1420 ' *** Sumatra ***                      [502]
1430 DATA 96,5,6,1,105,-5,106,-2,103,-1,10 [2703]
3,0,97,5,96,5,101,0
1440 ' *** Sachalin ***                     [440]
1450 DATA 143,47,4,0:' Init                 [643]
1460 DATA 143,54,145,48,143,49,143,47       [1713]
1470 ' *** Hokkaido ***                     [361]
1480 DATA 141,43,5,1:' Init                 [996]
1490 DATA 142,45,145,44,143,42,140,43,141, [1654]
43
1500 DATA 143,43:' Fill                     [818]
1510 ' *** Honshu ***                       [715]
1520 DATA 140,41,12,1:' Init                [942]
1530 DATA 141,41,142,40,141,36,136,33,132, [3802]
31,131,31,129,33,133,36,136,36,137,38,139,
38,140,41
1540 DATA 138,36:' Fill                     [767]
1550 ' *** GB ***                           [764]
1560 DATA -6,50,17,2:' Init                 [408]
1570 DATA 1,51,0,52,1,53,0,53,-3,56,-2,57. [2786]
5,-4,57.5,-3,58.5,-5,58.5,-6,56
1580 DATA -5,56,-5,55,-3,55,-3,53,-4,53,-5 [2667]
,52,-3,51
1590 DATA -1,52,-5,57:' Fill                [1198]
1600 ' *** Irland ***                       [1192]
1610 DATA -10,52,6,1:' Init                 [1102]
1620 DATA -6,52.5,-6,55,-8,55,-8,54,-10,54 [1567]
,-10,52
1630 DATA -8,53:' Fill                      [815]
1640 ' *** Sri Lanka ***                    [1202]
1650 DATA 80,9,3,1:' Init                   [805]
1660 DATA 80,6,82,7,80,9                    [525]
1670 DATA 81,8:' Fill                       [758]
1680 ' *** Tasmanien ***                    [641]
1690 DATA 148,-41,4,1:' Init                [745]
1700 DATA 145,-41,146,-43.5,147,-43.5,148, [1750]
-41
1710 DATA 147,-42:' Fill                    [957]
1720 ' *** Inseln ***                       [611]
1730 DATA -77,18:' Jamaika                  [663]
1740 DATA -66,18:' Puerto Rico              [1125]
1750 DATA -78,25:' Bahamas                  [1615]
1760 DATA -61,11:' Trinidad                 [957]
1770 DATA -63,18,-61,16,-61,14:' Karibik    [1707]
1780 DATA -60,-52,-59,-52:' Falkland        [1527]
1790 DATA -37,-54:' Suedgeorgien            [1554]
1800 DATA -90,0:' Galapagos                 [742]
1810 DATA -110,-29:' Osterinsel             [1587]
1820 DATA -9,72:' Jan Mayen                 [1540]
1830 DATA -7,62:' Foeroyar                  [672]
1840 DATA -2,60:' Shetland                  [1157]
1850 DATA 70,-50:' Kerguelen                [667]
1860 DATA 0,-160:' Jarvis                   [679]
1870 DATA 177,-18,179,-17:' Fiji            [1343]
1880 DATA -175,-21:' Tonga                  [645]
1890 DATA -178,-18:' Niue                   [1342]
1900 DATA -162,5,-160,4,-158,2:' Palmyra    [2060]
1910 DATA -177,28,-175,28,-170,25,-160,22, [3212]
-158,21,-157,20,-156,19:' Hawaii
1920 DATA -169,17:' Johnston                [635]
1930 DATA 166,19:' Wake                     [768]
1940 DATA 155,24:' Marcus                   [1121]
1950 DATA 167,-1:' Nauru                    [1147]
1960 DATA 161,10,162,11:' Eniwetok          [1158]
1970 DATA 166,11,167,11,166,9,169,8,170,12 [5218]
,170,10,172,6,173,2,175,-2:' Bikini,Marsha
ll und Gilbert
1980 DATA -145,16,-144,15:' Saipan,Guam     [1445]
1990 DATA 138,9,140,10:' Yap                [1122]
2000 DATA 135,8:' Palau                     [1345]
2010 DATA 146,-2,155,-6,157,-7,159,-8,162, [2439]
-11:' Bismarck
2020 DATA 166,-15,168,-18:' Vila            [1163]
2030 DATA -176,-44,179,-48,178,-50:' Chath [2212]
am-Antipode
```

Listing SAT

```
2040 DATA 7,0:' Sao Tome                        [1541]
2050 DATA 8,1:' Principe                        [540]
2060 DATA 40,-21:' Bassas                       [1360]
2070 DATA 55,-5,53,-6,46,-9,51,-9,56,-10,5 [3765]
5,-21,57,-20,63,-20,47,-12,43,-12:' Seyche
llen
2080 DATA 44,-13,55,-15,54,12,92,14,92,12, [3679]
92,10,93,7:' ind. Ozean
2090 DATA 120,-10,123,8,121,6,126,-3,129,- [3995]
3,130,-3,129,1,43,11,43,7,43,5
2100 DATA 43,3,43,1:' auch ind. Ozean und  [2296]
Phillipinen
2110 DATA 2,74:' Baeren                         [435]
2120 DATA 14,36:' Malta                         [1330]
2130 DATA 3,39,4,40:' Balearen                  [628]
2140 DATA -17,33:' Madeira                      [1287]
2150 DATA -17,27,-16,28,-15,28,-14,29:' Ka [1622]
naren
2160 DATA -23,6,-22,7,-23,8:' Kapverden         [2312]
2170 DATA -25,39,-23,38:' Azoren                [942]
2180 DATA -25,1:' St Paul                       [621]
2190 DATA -17,-3:' Ascension                    [687]
2200 DATA -13,-8:' St Helena                    [1333]
2210 DATA -17,-28,-15,-40:' Tristan da Cun [2613]
ha und Gough
2220 DATA -65,32:' Bermuda                      [389]
2230 DATA -108,-28:' Sala y Gomez               [891]
2240 DATA -142,-8,-141,-9,-139,-10:' Marqu [1718]
esas
2250 DATA -111,8:' Socorro                      [971]
2260 DATA -115,18:' Clarion                     [882]
2270 DATA -87,5:' Kokos                         [1097]
2280 DATA -82,4:' Malpelo                       [736]
2290 DATA 96,-12:' Tschagos                     [478]
2300 DATA -55,-61:' Suedshetlands               [1505]
2310 DATA -45,-61:' Suedorkney                  [1167]
2320 DATA -26,-59,-25,-57,-28,-56:' Suedsa [1824]
ndwich
2330 DATA -144,-27:' Rapa                       [682]
2340 DATA -173,61:' St.Martin                   [923]
2350 DATA 166,55,168,54:' Kommandoerins.        [2368]
2360 MODE 1                                     [506]
2370 ORIGIN 320,200                             [215]
2380 vx=1.76:vy=2.15:p0=0:p1=17:p2=2:p3=26 [2678]
2390 INK 0,p0:INK 1,p1:INK 3,p3:INK 2,p2        [1977]
2400 PAPER 0:CLS                                [973]
2410 'Gradnetz                                  [429]
2420 PEN 2:FOR i=-150 TO 150 STEP 30            [2466]
2430 PLOT i*vx,90*vy,2:DRAW i*vx,-90*vy,2       [1405]
2440 NEXT i                                     [375]
2450 FOR i=-60 TO 60 STEP 30                    [608]
2460 PLOT 180*vx,i*vy,2:DRAW -180*vx,i*vy, [2618]
2
2470 NEXT i                                     [375]
2480 'Kontinente umranden und fuellen           [3147]
2490 FOR k=1 TO 27                              [1101]
2500 READ ll,bb,h,hh:'init                      [1355]
2510 FOR i=1 TO h                               [446]
2520 ba=bb                                      [415]
2530 la=ll                                      [542]
2540 READ ll,bb                                 [1026]
2550 PLOT la*vx,ba*vy,1                         [1212]
2560 DRAW ll*vx,bb*vy,1                         [226]
2570 NEXT i                                     [375]
2580 IF hh=0 GOTO 2630                          [289]
2590 FOR i=1 TO hh                              [565]
2600 READ ll,bb                                 [1026]
2610 MOVE ll*vx,bb*vy:FILL 1                    [894]
2620 NEXT i                                     [375]
2630 NEXT k                                     [373]
2640 'Inseln punkten                            [882]
2650 FOR i=1 TO 131                             [531]
2660 READ ll,bb                                 [1026]
2670 PLOT ll*vx,bb*vy,1                         [1106]
2680 NEXT i                                     [375]
2690 'Rand                                      [401]
2700 PLOT -180*vx,90*vy,3                       [1115]
2710 DRAW -180*vx,-90*vy,3                      [748]
2720 DRAW 180*vx,-90*vy,3                       [428]
2730 DRAW 180*vx,90*vy,3                        [642]
2740 DRAW -180*vx,90*vy,3                       [914]
2750 a$=INKEY$                                  [278]
2760 IF a$="" GOTO 2750                         [657]
2770 IF a$="s" OR a$="S" THEN SAVE "sat.sc [3291]
r",b,&C000,&4000
```

Listing SAT

```
2780 MODE 2                                     [513]
2790 PEN 1                                      [549]
2800 BORDER 0                                   [1008]
2810 END                                        [110]
```

```
10 flag=0:'Hauptmenue                           [1345]
20 INK 0,0:CLS:LOCATE 1,2:PRINT " ====== [11902]
S A T - M A S T E R  ======":LOCATE 14,25
:PRINT CHR$(164);" DB1HZ 1986":LOCATE 9,7:
PRINT "Bahnberechnung        : 1":LOCATE 9,9:
PRINT "Satellit einf}gen     : 2"
30 LOCATE 9,11:PRINT "Satellit l¦schen     : [15193]
 3":LOCATE 9,13:PRINT "Bahndaten {ndern
: 4":LOCATE 9,15:PRINT "Satellit umnennen
 : 5":LOCATE 9,17:PRINT "Standort {ndern
  : 6":LOCATE 9,20:PRINT "Programmende
    : 0"
40 WHILE INKEY$<>"":WEND                        [1786]
50 a$=INKEY$:IF a$="" GOTO 50 ELSE IF a$=" [2705]
0" THEN 70
60 ON INSTR("123456",a$) GOSUB 1800,220,16 [3321]
30,870,550,3830:GOTO 20
70 'Programmende                                [1226]
80 IF flag=0 THEN CALL 0                        [574]
90 OPENOUT "sat.dat":WRITE #9,sat,meml,mem [12931]
b:FOR i=1 TO sat:WRITE #9,sat$(i),zi(i),zo
(i),ze(i),zw(i),zmm(i),zemm(i),zg(i),zumla
uf(i),zbake(i):NEXT:CLOSEOUT:¦ERA,"sat.bak
":CALL 0
100 'Submenue                                   [603]
110 CLS:IF sat=0 THEN 160                       [798]
120 LOCATE 9,1:PRINT "Welchen Satelliten ? [6600]
":FOR i=1 TO sat:LOCATE 7,3+i:PRINT sat$(i
);" :";:IF i<10 THEN PRINT i ELSE PRINT "
";CHR$(i+55)
130 NEXT i:LOCATE 7,4+i:PRINT "Hauptmenue  [2728]
      : 0"
140 j$=INKEY$:IF j$="" GOTO 140                 [2533]
150 IF j$="0" THEN RETURN ELSE j=INSTR(max [6496]
sat$,UPPER$(j$)):IF j=0 OR j>sat GOTO 140
ELSE RETURN
160 CLS:LOCATE 5,10:PRINT "Die Bahndaten s [5865]
ind fehlerhaft.":GOTO 210
170 'sat$ ausgeben                              [1713]
180 CLS:LOCATE 9,2:PRINT "Die Datei enth{l [3952]
t:":IF sat=0 THEN 160
190 FOR i=1 TO sat:LOCATE 3,4+i:PRINT sat$ [5835]
(i);"  :";:IF i<10 THEN PRINT i ELSE PRINT
 " ";CHR$(i+55)
200 NEXT                                        [350]
210 LOCATE 39,25:PRINT ">";:CALL &BB18:RET [1174]
URN
220 'Satellit eingeben                          [672]
230 IF sat<maxsat THEN 250                      [1334]
240 CLS:LOCATE 10,10:PRINT "Datei voll !": [3313]
GOTO 210
250 GOSUB 660                                   [881]
260 IF a$="0" THEN RETURN                       [1402]
270 IF sat=0 GOTO 310                           [1229]
280 i=1                                         [423]
290 IF UPPER$(name$)=UPPER$(sat$(i)) GOTO  [1867]
360
300 IF i<sat THEN i=i+1:GOTO 290                [1934]
310 sat=sat+1                                   [307]
320 sat$(sat)=name$                             [1344]
330 j=sat                                       [639]
340 flag=1                                      [262]
350 GOTO 900                                    [320]
360 'Satellit doppelt                           [385]
370 CLS                                         [91]
380 LOCATE 14,8                                 [689]
390 PRINT "Der Satellit"                        [2610]
400 LOCATE 15,10                                [609]
410 PRINT "'";name$;"'"                         [737]
420 LOCATE 9,12                                 [542]
430 PRINT "ist bereits vorhanden."              [3083]
440 LOCATE 10,17                                [689]
450 PRINT "]berschreiben(J/N) ?"                [2336]
460 k$=UPPER$(INKEY$):IF INKEY$="" THEN 46 [1448]
0
470 IF k$="N" THEN RETURN                       [1033]
```

Listing SAT

```
480 IF k$="J" GOTO 510                          [1033]
490 IF k$="O" THEN RETURN                       [971]
500 GOTO 460                                    [456]
510 'Namen ueberschreiben                       [773]
520 sat$(i)=name$                               [1161]
530 j=i                                         [483]
540 GOTO 900                                    [320]
550 'Satnamen {ndern                            [1205]
560 GOSUB 100:'alten Namen erfragen             [2408]
570 IF j$="O" THEN RETURN                       [823]
580 CLS:name$="":a$=""                          [989]
590 LOCATE 7,8                                  [321]
600 PRINT "Neuer Name des Satelliten :"         [3674]
610 GOSUB 710:'Neuen Namen erfragen             [2628]
620 sat$(j)=name$                               [1063]
630 flag=1                                      [262]
640 GOSUB 170:'Datei ausgeben                   [1714]
650 RETURN                                      [555]
660 'Satellitennamen eingeben                   [661]
670 name$=""                                    [404]
680 CLS                                         [91]
690 LOCATE 10,8                                 [515]
700 PRINT "Name des Satelliten :"               [3275]
710 LOCATE 10,11                                [679]
720 PRINT ">                  <"                [1440]
730 LOCATE 11,11                                [512]
740 a$=INKEY$:IF a$="" GOTO 740                 [1495]
750 IF a$="O" AND LEN(name$)=0 THEN RETURN      [1769]
760 IF a$=CHR$(13) GOTO 840                     [793]
770 IF LEN(name$)=18 AND a$<>CHR$(242) GOT      [2246]
O 850
780 IF a$=CHR$(243) THEN a$=" "                 [762]
790 IF a$<>CHR$(242) OR name$="" GOTO 830       [1775]
800 name$=LEFT$(name$,(LEN(name$)-1))           [1359]
810 LOCATE 11+LEN(name$),11:PRINT " "           [855]
820 LOCATE 11+LEN(name$),11:GOTO 740            [781]
830 IF LEN(name$)<20 AND a$<>CHR$(13) THEN      [5312]
 name$=name$+a$:PRINT a$;:GOTO 740
840 IF LEN(name$)<20 THEN name$=name$+" ":      [1676]
GOTO 840
850 a$=""                                       [388]
860 RETURN                                      [555]
870 'Bahndaten {ndern                           [2826]
880 GOSUB 100:'Submenue                         [1022]
890 IF j$="O" THEN RETURN                       [823]
900 'Bahndateneingabe                           [1879]
910 CLS                                         [91]
920 LOCATE 6,1                                  [336]
930 PRINT "Satellit ";sat$(j)                   [1794]
940 LOCATE 1,4                                  [620]
950 h=0                                         [416]
960 PRINT "Inklination                 : ";     [1664]
970 IF zi(j)>0 THEN PRINT zi(j):LOCATE 25,      [3136]
4
980 INPUT h                                     [448]
990 IF h<>0 THEN zi(j)=h:h=0:flag=1             [2542]
1000 LOCATE 1,6                                 [626]
1010 PRINT "Nutation (RAAN)            : ";     [2945]
1020 IF zo(j)<>0 THEN PRINT zo(j):LOCATE 2      [3487]
5,6
1030 INPUT h                                    [448]
1040 IF h<>0 THEN zo(j)=h:h=0:flag=1            [2010]
1050 LOCATE 1,8                                 [600]
1060 PRINT "Exzentrizit{t              : ";     [2547]
1070 IF ze(j)<>0 THEN PRINT ze(j);:LOCATE       [2870]
25,8
1080 INPUT h                                    [448]
1090 IF h<>0 THEN ze(j)=h:h=0:flag=1            [1750]
1100 LOCATE 1,10                                [726]
1110 PRINT "Argument des Perig{ums : ";         [2551]
1120 IF zw(j)<>0 THEN PRINT zw(j);:LOCATE       [1763]
25,10
1130 INPUT h                                    [448]
1140 IF h<>0 THEN zw(j)=h:flag=1                [2062]
1150 h=0                                        [416]
1160 LOCATE 1,12                                [724]
1170 PRINT "Mittlere Bewegung (MM) : ";         [2794]
1180 IF zmm(j)<>0 THEN PRINT zmm(j);:LOCAT      [2676]
E 25,12
1190 INPUT h                                    [448]
1200 IF h<>0 THEN zmm(j)=h:flag=1               [2036]
1210 h=0                                        [416]
1220 LOCATE 1,14                                [738]
1230 PRINT "Abbremsung (EMM)           : ";     [3829]
```

Listing SAT

```
1240 IF zemm(j)<>0 THEN PRINT zemm(j);:LOC      [4121]
ATE 25,14
1250 INPUT h                                    [448]
1260 IF h<>0 THEN zemm(j)=h:flag=1              [2205]
1270 h=0                                        [416]
1280 LOCATE 1,16                                [736]
1290 PRINT "Epoche Jahr                : ";     [2183]
1300 INPUT h                                    [448]
1310 IF h=0 THEN 1430 ELSE hj=h                 [1596]
1320 LOCATE 1,18                                [750]
1330 PRINT "Epoche Tag                 : ";     [1668]
1340 INPUT ht                                   [381]
1350 IF ht<0 OR ht>366 THEN 1340                [1080]
1360 LOCATE 1,20                                [732]
1370 PRINT "Referenzanomalie (MA)      : ";     [2441]
1380 INPUT h                                    [448]
1390 IF h<0 THEN 1380                           [1710]
1400 hj=hj-85:IF hj>1000 THEN hj=hj-1900        [2046]
1410 zg(j)=ht+2556+INT(365.25*hj)-h/(360*z      [1981]
mm(j))
1420 flag=1:ht=4:GOTO 1490                      [1712]
1430 h=0:ht=0                                   [668]
1440 LOCATE 1,16                                [736]
1450 PRINT "Referenzperig{um           : ";     [3298]
1460 IF zg(j)<>0 THEN PRINT zg(j);:LOCATE       [2408]
25,16
1470 INPUT h                                    [448]
1480 IF h<>0 THEN zg(j)=h:flag=1                [2013]
1490 h=0                                        [416]
1500 LOCATE 1,18+ht                             [886]
1510 PRINT "Referenz-Umlaufnummer      : ";     [3288]
1520 IF zumlauf(j)<>0 THEN PRINT zumlauf(j      [3446]
);:LOCATE 25,18+ht
1530 INPUT h                                    [448]
1540 IF h<>0 THEN zumlauf(j)=h:flag=1           [2348]
1550 h=0                                        [416]
1560 LOCATE 1,20+ht                             [965]
1570 PRINT "Bakenfrequenz in MHz       : ";     [3230]
1580 IF zbake(j)<>0 THEN PRINT zbake(j);:L      [2426]
OCATE 25,20+ht
1590 INPUT h                                    [448]
1600 IF h<>0 THEN zbake(j)=h:flag=1             [3153]
1610 GOSUB 170:'Datei ausgeben                  [1714]
1620 RETURN                                     [555]
1630 'Satellit loeschen                         [874]
1640 GOSUB 100                                  [929]
1650 IF j$="O" THEN RETURN                      [823]
1660 IF j=sat THEN 1710                         [1079]
1670 FOR i=j TO sat-1                           [1622]
1680 sat$(i)=sat$(i+1):zi(i)=zi(i+1):zo(i)      [10294]
=zo(i+1):ze(i)=ze(i+1):zw(i)=zw(i+1):zmm(i
)=zmm(i+1):zemm(i)=zemm(i+1):zg(i)=zg(i+1)
:zumlauf(i)=zumlauf(i+1):zbake(i)=zbake(i+
1)
1690 NEXT i                                     [375]
1700 sat$(sat)="":zi(sat)=0:zo(sat)=0:ze(s      [9381]
at)=0:zw(sat)=0:zmm(sat)=0:zemm(sat)=0:zg(
sat)=0:zumlauf(sat)=0:zbake(sat)=0
1710 sat=sat-1                                  [319]
1720 flag=1                                     [262]
1730 GOSUB 170:'Datei ausgeben                  [1714]
1740 RETURN                                     [555]
1750 'Fehler                                    [443]
1760 CLS                                        [91]
1770 LOCATE 10,10                               [676]
1780 PRINT "Bahndaten fehlerhaft."              [2489]
1790 RETURN                                     [555]
1800 'Eingabe/Berechnung                        [1613]
1810 GOSUB 100                                  [929]
1820 IF j$="O" THEN RETURN                      [823]
1830 ON ERROR GOTO 1750                         [1341]
1840 'Bahndatenadaption                         [2449]
1850 i=zi(j):o=zo(j):e=ze(j):w=zw(j):mm=zm      [5290]
m(j):emm=zemm(j):g=zg(j):umlauf=zumlauf(j)
:zbake=zbake(j)
1860 sat$=sat$(j):l=meml:b=memb                 [1558]
1870 IF i=0 OR o=0 OR w=0 OR mm=0 OR g=0 T      [1310]
HEN 1750
1880 'Datumeingabe                              [836]
1890 CLS                                        [91]
1900 LOCATE 8,3                                 [326]
1910 PRINT "Satellit ";sat$                     [1743]
1920 LOCATE 8,6                                 [335]
1930 INPUT "Tag     : ",h                       [1243]
```

Listing SAT

```
1940 IF h<>0 THEN t=h:h=7 ELSE t=memt:GOTO [3466]
 2110
1950 IF t<1 OR t>31 GOTO 1920                [1361]
1960 rt=t                                    [249]
1970 LOCATE 8,8                              [317]
1980 INPUT "Monat  : ",m                     [1235]
1990 IF (t>29 AND m=2) OR (t>30 AND (m=4 O [1939]
R m=6 OR m=9 OR m=11)) OR m<1 OR m>12 GOTO
 1970
2000 rm=m                                    [583]
2010 LOCATE 8,10                             [544]
2020 INPUT "Jahr    : 19",j                  [932]
2030 IF j<0 OR j>=100 GOTO 2010              [1310]
2040 rj=j                                    [554]
2050 m=m+1                                   [200]
2060 IF m<4 THEN j=j-1:m=m+12                [915]
2070 t=INT(j*365.25)+INT(m*30.6)+t-28553     [1639]
2080 LOCATE 8,8+h                            [390]
2090 INPUT "UTC      : ",u                   [1093]
2100 IF u<0 OR u>24 THEN 2080                [582]
2110 'Bahnberechnung                         [1336]
2120 mm=mm+emm*(t-g)/2                       [981]
2130 n=2*PI*mm:pev=0                         [1503]
2140 a=42220*mm^(-2/3)/6378                  [1190]
2150 z=a^3.5*(1-e*e)^2                       [1376]
2160 q=-9.98*COS(i)/z                        [1069]
2170 v=(25*(COS(i))^2-5)/z                   [1622]
2180 ON ERROR GOTO 0                         [1381]
2190 INK 0,p0:PEN 3,1                        [1058]
2200 reft=TIME/300                           [1054]
2210 ORIGIN 320,200                          [215]
2220 |COPY,&5600,&4000,&C000                 [1401]
2230 IF memt=t THEN 2260                     [1193]
2240 u=INT(u)+(u-INT(u))*5/3                 [1239]
2250 t=u/24+0.00001+INT(t)                   [751]
2260 u=24*(t-INT(t))                         [581]
2270 pt=INT(t)                               [545]
2280 pu=INT(u)+INT((u-INT(u))*300/5)/100     [1543]
2290 d=t-g                                   [599]
2300 k=d*q+o-100.3-t*f-1                     [1258]
2310 m=d*n                                   [172]
2320 RAD                                     [301]
2330 r=e                                     [104]
2340 p=m                                     [120]
2350 GOSUB 3600                              [1192]
2360 h=(m-p+y)/(1-x)                         [1161]
2370 p=p+h                                   [311]
2380 IF ABS(h)>0.000015 GOTO 2350            [1114]
2390 r=1                                     [252]
2400 GOSUB 3600                              [1192]
2410 y=y*SQR(1-e*e)                          [927]
2420 x=x-e                                   [1037]
2430 DEG                                     [133]
2440 GOSUB 3610                              [1198]
2450 r=a*r                                   [283]
2460 p=p+w+d*v                               [798]
2470 GOSUB 3600                              [1192]
2480 h=x                                     [504]
2490 p=i                                     [108]
2500 r=y                                     [116]
2510 GOSUB 3600                              [1192]
2520 s=y                                     [152]
2530 y=x                                     [571]
2540 x=h                                     [551]
2550 GOSUB 3610                              [1198]
2560 p=p+k                                   [308]
2570 k=p+l                                   [245]
2580 u=r                                     [115]
2590 GOSUB 3600                              [1192]
2600 z=s-SIN(b)                              [1155]
2610 x=x-COS(b)                              [1349]
2620 h=y                                     [507]
2630 y=z                                     [577]
2640 GOSUB 3610                              [1198]
2650 p=p-b+90                                [411]
2660 GOSUB 3600                              [1192]
2670 j=y                                     [531]
2680 y=h                                     [523]
2690 GOSUB 3610                              [1198]
2700 c=180-p                                 [721]
2710 x=r                                     [573]
2720 y=j                                     [529]
2730 GOSUB 3610                              [1198]
2740 m=m/(2*PI)                              [82]
2750 puml=INT(m)+umlauf                      [1274]
```

Listing SAT

```
2760 pc=ROUND(c)                             [456]
2770 pp=ROUND(p)                             [653]
2780 pm=256*(m-INT(m))                       [860]
2790 ' SSP-Berechnung                        [1391]
2800 k=k/360                                 [458]
2810 k=360*(k-INT(k))                        [605]
2820 x=u                                     [570]
2830 y=s                                     [584]
2840 GOSUB 3610                              [1198]
2850 IF k>180 THEN k=k-360                   [2498]
2860 ' Sichtbereich                          [1574]
2870 RAD                                     [301]
2880 s1=SQR(r*r-1)/r                         [1467]
2890 s1=ATN(s1/SQR(1-s1*s1)):'ARCSIN         [2740]
2900 s2=PI/180*(90-p)                        [1478]
2910 s3=s2-s1                                [1157]
2920 f1=0:IF s3<0 THEN s3=s1-s2:f1=1         [1212]
2930 IF pp<-5 THEN 3050                      [612]
2940 'Entfernung                             [1423]
2950 DEG                                     [133]
2960 dx=r*SIN(k)-SIN(l):dy=r*COS(k)-COS(l) [3172]
:dz=r*SIN(p)-SIN(b)
2970 entf=6378*SQR(dx*dx+dy*dy+dz*dz)        [1476]
2980 RAD                                     [301]
2990 'Doppler                                [1095]
3000 IF pev=0 THEN dop=zbake:GOTO 3040       [1534]
3010 det=(t-det)*86400                       [1070]
3020 dop=(pev-entf)/det                      [536]
3030 dop=ROUND(zbake*2.99792458E+09/(29979 [3564]
2.458-dop))/10000
3040 pev=entf:entf=ROUND(entf):det=t         [3138]
3050 ' Ausgabe                               [572]
3060 |COPY,&5600,&4000,&C000                 [1401]
3070 PEN 3,1                                 [807]
3080 LOCATE 2,2:PRINT sat$;                  [1519]
3090 rm$=STR$(rm)                            [473]
3100 datum$=STR$(rt)+"."+RIGHT$(rm$,LEN(rm [3889]
$)-1)+".19"+RIGHT$(STR$(rj),2)
3110 LOCATE 40-LEN(datum$),2:PRINT datum$; [3887]
3120 IF pp<-5 THEN 3160                      [606]
3130 LOCATE 2,23:PRINT "Entfernung";entf;" [4537]
km";
3140 IF dop=0 THEN 3160 ELSE dop$="Bake"+S [4042]
TR$(dop)+" MHz"
3150 LOCATE 40-LEN(dop$),23:PRINT dop$;      [2795]
3160 LOCATE 1,25                             [731]
3170 PRINT USING "##.## UTC ";pu;            [2059]
3180 PRINT USING "###/";pc;                 [1206]
3190 PRINT USING "###";pp;                  [922]
3200 PRINT USING " MA ### ";pm;              [1391]
3210 PRINT USING " Umlauf #####";puml;       [2259]
3220 LOCATE 1,25                             [731]
3230 PLOT k*1.76-5,p*2.15-4,3:DRAW k*1.76+ [4545]
4,p*2.15+4,3:PLOT k*1.76-5,p*2.15+4,3:DRAW
 k*1.76+4,p*2.15-4,3
3240 s3=s3+PI/96 75                          [1260]
3250 hb=90-s3*180/PI:IF hb>75 OR f1=0 THEN [2608]
 3290 ELSE hb=76
3260 PLOT -180*1.76,hb*2.15,3:DRAW 180*1.7 [2318]
6,hb*2.15,3
3270 hb=hb-8/2.15                            [280]
3280 IF hb<90-s3*180/PI THEN s3=PI/180*(90 [2632]
-hb) ELSE 3260
3290 hb=90-s3*180/PI:IF hb>78 THEN 3410      [1812]
3300 IF hb<0 THEN hb=hb-0.465:'Nullsymmetr [2368]
ie
3310 c1=(COS(s1)-COS(s2)*COS(s3))/(SIN(s2) [3281]
*SIN(s3)):'Seitencosinussatz
3320 IF pp>-6 AND hb<-70 THEN 3420           [567]
3330 IF c1>1 OR hb<-83 THEN 3420             [1554]
3340 IF c1<-1 THEN c1=180:GOTO 3360          [1217]
3350 c1=90-180/PI*ATN(c1/SQR(1-c1^2)):'ARC [3552]
COS
3360 hl1=k-c1:hl2=k+c1:hb=hb*2.15            [1469]
3370 IF hl1<=-180 AND hl2>=180 THEN PLOT - [3520]
180*1.76,hb,3:DRAW 180*1.76,hb,3:GOTO 3410
3380 IF hl1<-180 THEN PLOT -180*1.76,hb,3: [6294]
DRAW hl2*1.76,hb,3:PLOT (hl1+360)*1.76,hb,
3:DRAW 180*1.76,hb,3:GOTO 3410
3390 IF hl2>180 THEN PLOT -180*1.76,hb,3:D [4896]
RAW (hl2-360)*1.76,hb,3:PLOT hl1*1.76,hb,3
:DRAW 180*1.76,hb,3:GOTO 3410
3400 PLOT hl1*1.76,hb,3:DRAW hl2*1.76,hb,3 [2089]
3410 s3=s3+PI/180*8/2.15:GOTO 3290           [1819]
3420 DEG                                     [133]
```

Listing SAT

```
3430 a$=INKEY$                                    [278]
3440 IF (reft-TIME/600)-INT(reft-TIME/600)       [3957]
<0.5 THEN kk=3 ELSE kk=2
3450 PLOT k*1.76-5,p*2.15-4,kk:DRAW k*1.76       [3505]
+4,p*2.15+4,kk:PLOT k*1.76-5,p*2.15+4,kk:D
RAW k*1.76+4,p*2.15-4,kk
3460 IF a$="0" THEN memt=t:PEN 3,0:ON ERRO       [1876]
R GOTO 0:CLS:RETURN
3470 IF a$=CHR$(240) OR a$=CHR$(241) OR a$       [2262]
=CHR$(242) OR a$=CHR$(243) THEN 3530
3480 IF TIME/300-reft<30 GOTO 3430                [847]
3490 t=t+1/2880                                   [350]
3500 IF t-INT(t)<(t-1/2880)-INT(t-1/2880)        [2854]
THEN GOSUB 4360
3510 reft=reft+30                                 [781]
3520 GOTO 2260                                    [359]
3530 IF a$=CHR$(240) THEN t=t-1/144:IF t-I       [4851]
NT(t)>(t+1/144)-INT(t+1/144) THEN GOSUB 44
80
3540 IF a$=CHR$(241) THEN t=t+1/144:IF t-I       [3937]
NT(t)<(t-1/144)-INT(t-1/144) THEN GOSUB 43
60
3550 IF a$=CHR$(242) THEN t=t-1/1440:IF t-       [3271]
INT(t)>(t+1/1440)-INT(t+1/1440) THEN GOSUB
 4480
3560 IF a$=CHR$(243) THEN t=t+1/1440:IF t-       [4198]
INT(t)<(t-1/1440)-INT(t-1/1440) THEN GOSUB
 4360
3570 a$=INKEY$:IF a$<>"" THEN 3460               [2012]
3580 reft=TIME/300                               [1054]
3590 GOTO 2260                                    [359]
3600 y=r*SIN(p):x=r*COS(p):RETURN                [1564]
3610 r=SQR(x*x+y*y):IF x=0 THEN p=90*SGN(y       [1623]
):RETURN
3620 p=ATN(y/x):IF x>0 THEN RETURN               [2284]
3630 IF y>0 THEN p=p+180:RETURN                  [1161]
3640 p=p-180:RETURN                               [696]
3650 'Satellit aendern                            [416]
3660 GOSUB 100                                    [929]
3670 IF j$="0" OR sat=0 THEN RETURN              [1978]
3680 'Eingabe Satellitenname                     [2450]
3690 name$=""                                     [404]
3700 CLS                                           [91]
3710 LOCATE 8,8                                   [317]
3720 PRINT "Name des Satelliten :"               [3275]
3730 LOCATE 12,11                                 [472]
3740 a$=INKEY$:IF a$="" GOTO 3740                [1579]
3750 IF a$=CHR$(13) GOTO 3810                     [572]
3760 IF LEN(name$)=0 AND a$="0" THEN RETUR       [1716]
N
3770 IF a$=CHR$(242) THEN name$=LEFT$(name       [7262]
$,LEN(name$)-1):LOCATE 12+LEN(name$),12:PR
INT " ":LOCATE 12+LEN(name$),12:GOTO 3740
3780 IF a$=CHR$(243) THEN a$=" "                  [762]
3790 IF LEN(name$)<10 THEN name$=name$+a$        [2665]
ELSE GOTO 3820
3800 PRINT a$;:GOTO 3740                         [1412]
3810 IF LEN(name$)<10 THEN name$=name$+" "       [3055]
:GOTO 3810
3820 sat$(j)=name$:GOTO 900                      [1554]
3830 'QTH abspeichern                            [1188]
3840 CLS:LOCATE 6,2:PRINT "===== Standort       [18112]
{ndern =====":LOCATE 4,5:PRINT"Die Eingabe
 soll erfolgen als :":LOCATE 8,10:PRINT "L
{nge/Breite           : 1":LOCATE 8,12:PRINT
"QTH-Locator (neu)    : 2":LOCATE 8,15:PRIN
T "Hauptmenue             : 0"
3850 a$=INKEY$:IF a$="" GOTO 3850                [1730]
3860 IF a$="0" THEN RETURN ELSE a=VAL(a$):       [4261]
ON a GOTO 3870,3930:GOTO 3850
3870 'Locator: L/B                               [1355]
3880 CLS                                           [91]
3890 LOCATE 10,10:PRINT "L{nge  :                [5655]
 Grad Ost":LOCATE 19,10:INPUT l:IF l<-180
OR l>180 THEN 3890
3900 l=INT(l)+5/3*(l-INT(l))                     [1976]
3910 LOCATE 10,12:PRINT "Breite :                [5480]
 Grad Nord":LOCATE 19,12:INPUT b:IF b<-90
OR b>90 THEN 3910
3920 b=INT(b)+5/3*(b-INT(b)):GOTO 4310           [2106]
3930 'QTH-Locator                                [1316]
3940 CLS:a$="":name$="":LOCATE 8,10:PRINT        [2101]
"Locator : >          <":LOCATE 19,10
3950 a$=UPPER$(INKEY$):IF a$="" GOTO 3950        [2071]
```

Listing SAT

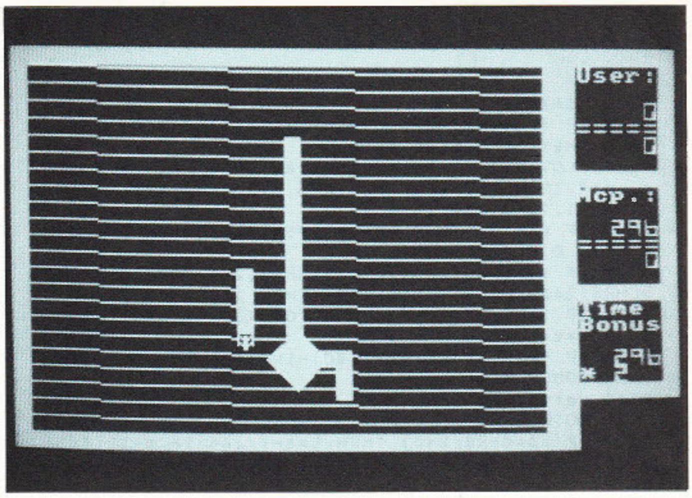
```
3960 IF LEN(name$)=0 AND a$="0" THEN RETUR       [1716]
N
3970 IF a$=CHR$(13) AND LEN(name$)<6 THEN        [2350]
RETURN
3980 IF (a$=CHR$(242) OR a$=CHR$(127)) AND       [9237]
 LEN(name$)>0 THEN name$=LEFT$(name$,LEN(n
ame$)-1):LOCATE 19,10:PRINT name$;" ";:LOC
ATE 19+LEN(name$),10:GOTO 3950
3990 IF LEN(name$)=6 THEN 4050                   [1567]
4000 IF (LEN(name$)<2 OR LEN(name$)>3) AND       [4629]
 (ASC(a$)<64 OR ASC(a$)>90) THEN 3950
4010 IF (LEN(name$)=2 OR LEN(name$)=3) AND       [5137]
 (ASC(a$)<49 OR ASC(a$)>57) THEN 3950
4020 name$=name$+a$                               [388]
4030 PRINT a$;                                    [420]
4040 GOTO 3950                                    [483]
4050 FOR k=1 TO 6                                [1080]
4060 u$=MID$(name$,k,1)                           [883]
4070 IF k<3 OR k>4 THEN u(k)=ASC(u$)-65 EL       [1950]
SE u(k)=VAL(u$)
4080 NEXT k                                       [373]
4090 l=u(1)*20+u(3)*2+(u(5)-0.5)/12-180          [1163]
4100 IF l<0 THEN l=l+360                         [1013]
4110 b=u(2)*10+u(4)+(u(6)-0.5)/24-90             [2275]
4120 LOCATE 8,13                                  [553]
4130 PRINT "--> ";                                [735]
4140 l(1)=INT(l)                                  [988]
4150 l(2)=60*(l-l(1))                             [703]
4160 l(3)=(l(2)-INT(l(2)))*3/5                   [1588]
4170 l(2)=INT(l(2))                              [1143]
4180 PRINT USING "###";l(1);                      [761]
4190 PRINT "#";USING "##";l(2);                  [1268]
4200 PRINT "'";USING "##";l(3);                   [505]
4210 PRINT CHR$(34);" Ost"                       [1723]
4220 LOCATE 12,15                                 [484]
4230 b(1)=INT(b)                                 [1128]
4240 b(2)=60*(b-b(1))                            [1090]
4250 b(3)=(b(2)-INT(b(2)))*3/5                   [1429]
4260 b(2)=INT(b(2))                               [928]
4270 PRINT USING "###";b(1);                     [1064]
4280 PRINT "#";USING "##";b(2);                  [1045]
4290 PRINT "'";USING "##";b(3);                   [808]
4300 PRINT CHR$(34);" Nord"                      [1585]
4310 LOCATE 13,20                                 [682]
4320 PRINT "Speichern ?"                         [1493]
4330 a$=UPPER$(INKEY$):IF a$="" THEN 4330        [1878]
4340 IF a$="J" OR a$="Y" THEN memb=b:meml=       [2363]
l:flag=1
4350 RETURN                                       [555]
4360 ' Datumaktualisierung aufw{rts              [1669]
4370 mm=mm+emm                                    [352]
4380 IF (rm=1 OR rm=3 OR rm=5 OR rm=7 OR r       [4250]
m=8 OR rm=10) AND rt=31 THEN rt=1:rm=rm+1:
RETURN
4390 IF rm=12 AND rt=31 THEN rt=1:rm=1:rj=       [1400]
rj+1:RETURN
4400 IF rm<>2 AND rt=30 THEN rt=1:rm=rm+1:       [2449]
RETURN
4410 IF rj/2000-INT(rj/2000)=0 AND rm=2 AN       [3793]
D rt=28 THEN rt=1:rm=3:RETURN
4420 IF rj/400-INT(rj/400)=0 AND rm=2 AND        [4890]
rt=29 THEN rt=1:rm=3:RETURN
4430 IF rj/100-INT(rj/100)=0 AND rm=2 AND        [2700]
rt=28 THEN rt=1:rm=3:RETURN
4440 IF rj/4-INT(rj/4)=0 AND rm=2 AND rt=2       [4637]
9 THEN rt=1:rm=3:RETURN
4450 IF rm=2 AND rt=28 THEN rt=1:rm=3:RETU       [2632]
RN
4460 rt=rt+1                                      [267]
4470 RETURN                                       [555]
4480 ' Datumaktualisierung abw{rts               [1714]
4490 rt=rt-1:mm=mm-emm                            [827]
4500 IF rt>0 THEN RETURN                         [1554]
4510 IF rm=2 OR rm=4 OR rm=6 OR rm=8 OR rm       [2677]
=9 OR rm=11 THEN rt=31:rm=rm-1:RETURN
4520 IF rm=1 THEN rt=31:rm=12:rj=rj-1:RETU       [4556]
RN
4530 IF rm<>3 THEN rt=30:rm=rm-1:RETURN          [1528]
4540 rm=2:rt=28                                   [323]
4550 IF rj/4-INT(rj/4)=0 THEN rt=29              [1364]
4560 IF rj/100-INT(rj/100)=0 THEN rt=28          [1268]
4570 IF rj/400-INT(rj/400)=0 THEN rt=29          [2219]
4580 IF rj/2000-INT(rj/2000)=0 THEN rt=28        [1505]
4590 RETURN                                       [555]
```

Listing SAT

Soweit die Reaktion reicht

Schnelles Reaktionsspiel für zwei Personen

Dieses Spiel basiert auf dem Film "Tron", der für großes Aufsehen gesorgt hatte. Umsetzungen wurden schon für die verschiedensten Computer geschrieben. Diese Version bietet aber einiges an Abwechslungen, die das Spielgeschehen um vieles spannender gestaltet.



Schnelle Entscheidungen werden bei Tron gefordert

Die Story

Sie befinden sich in einem Computer, der durch seine Intelligenz eine Gefahr geworden ist. Nun müssen Sie schleunigst in diesen Computer eindringen und das Hauptprogramm zerstören. Kaum sind Sie in das System eingedrungen, beginnt ein Abwehrprogramm (der zweite Spieler), Sie zu behindern und aus dem System zu werfen.

Aufgabe

Spieler eins muß mit seinem Joystick einen kleinen Gleiter über ein Spielfeld steuern, der eine unüberwindbare Linie hinterläßt. Spieler zwei muß nun mit seinem Gleiter und seinen Cursortasten versuchen, Spieler eins von seinem Weg abzubringen, indem er eine Energiespur hinterläßt – dies geschieht automatisch, wie bei Spieler eins.

Wird nun eine der Linien gekreuzt, explodiert der Gleiter, und eine Runde ist verloren. Wenn die zehnte Runde gespielt wurde – und das ist nicht leicht, da sich der Hintergrund in jedem Level ändert –, gibt es eine Auswertung der gespielten Runden.

Weitere Hinweise

Die Positionen der Gleiter werden per Zufall gesetzt, also aufgepaßt. Vor jedem Start werden sechs Piepsgeräusche ausgegeben, die den Spielstart signalisieren. Vom Titelbild aus startet man das Rennen mit dem "Feuerknopf" oder der SPACE-Taste. (Auf die gleiche Weise verläßt man den Info-Screen, der beim Abschluß des gesamten Rennens erscheint).

(Gregor Bileck/cd)

für 464-664-6128

(464 mit Emulator)

```
10 '************              [502]
20 '*   TRON   *              [539]
30 '************              [502]
40 '                          [117]
50 '                          [117]
60 CLEAR                      [217]
70 '                          [117]
80 '                          [117]
90 '*** Sounds                [453]
100 '                         [117]
110 ENV 1,127,-1,7            [496]
120 ENV 2,127,-1,40           [867]
130 ENV 3,127,-1,80           [449]
140 ENV 4,127,-1,1            [744]
150 ENT -1,5,-5,1,5,5,1       [721]
160 ENT -2,5,-20,1,5,15,1     [1194]
170 '                         [117]
180 '*** Symbols              [799]
190 '                         [117]
200 '---> Numbers             [802]
210 '                         [117]
220 SYMBOL AFTER 48           [1272]
230 SYMBOL 48,126,66,66,66,70,70,70,126   [1465]
240 SYMBOL 49,8,8,8,8,24,24,24,24         [1860]
250 SYMBOL 50,126,6,6,126,64,96,96,126    [1511]
260 SYMBOL 51,126,2,2,126,2,6,6,126       [2355]
270 SYMBOL 52,66,98,98,126,2,2,2,2        [2598]
280 SYMBOL 53,126,96,96,126,2,6,6,126     [1796]
290 SYMBOL 54,64,64,64,126,66,98,98,126   [2041]
300 SYMBOL 55,126,2,2,2,6,6,6,6           [1681]
310 SYMBOL 57,126,70,70,126,2,2,2,2       [1770]
320 SYMBOL 56,126,66,66,126,66,70,70,126  [2646]
330 '                         [117]
340 '---> glider              [657]
```

Listing TRON

```
350 '                                          [117]
360 SYMBOL 250,24,24,189,219,153,189,219,1 [3022]
29 '*** up
370 SYMBOL 251,129,219,189,153,219,189,24, [3231]
24 '*** down
380 SYMBOL 252,63,18,36,254,254,36,18,63 ' [2731]
*** left
390 SYMBOL 253,252,72,36,127,127,36,72,252 [3640]
 '*** right
400 '                                      [117]
410 '                                      [117]
420 w$=CHR$(143)                           [1447]
430 shu$=CHR$(250)                         [990]
440 shd$=CHR$(251)                         [965]
450 shl$=CHR$(252)                         [562]
460 shr$=CHR$(253)                         [1025]
470 '                                      [117]
480 '--------------------------------      [1611]
490 '                                      [117]
500 '**** Title ***                        [239]
510 '                                      [117]
520 MODE 1                                 [506]
530 BORDER 0                               [1008]
540 INK 0,0                                [53]
550 INK 1,0                                [49]
560 INK 2,0                                [109]
570 INK 3,0                                [105]
580 '                                      [117]
590 GRAPHICS PEN 3                         [947]
600 FOR oft=1 TO 100                       [729]
610 x=RND*640:y=RND*200+200                [2829]
620 PLOT x,y                               [471]
630 NEXT                                   [350]
640 '                                      [117]
650 GRAPHICS PEN 2                         [948]
660 y=200:st=2                             [802]
670 FOR oft=1 TO 20                        [690]
680 PLOT 0,y:DRAW 640,y                    [731]
690 y=y-st:st=st+2                         [1816]
700 NEXT                                   [350]
710 '                                      [117]
```

Listing TRON

```
720 GRAPHICS PEN 1                          [949]
730 x=320:y=200:gr=400                      [1071]
740 DEG                                     [133]
750 FOR a=115 TO 245 STEP 10                [1220]
760 PLOT 320,200                            [621]
770 DRAW x+gr*SIN(a),y+gr*COS(a)            [576]
780 NEXT                                    [350]
790 '                                       [117]
800 PLOT 80,360:DRAW 80,340:DRAW 100,340:D [5071]
RAW 120,260:DRAW 140,260:DRAW 160,340:DRAW
 180,340:DRAW 180,360:DRAW 80,360
810 PLOT 200,360:DRAW 240,360:DRAW 280,340 [2623]
:DRAW 240,320:DRAW 300,260:DRAW 200,260:DR
AW 200,360
820 PLOT 340,360:DRAW 400,360:DRAW 420,340 [6892]
:DRAW 420,280:DRAW 400,260:DRAW 340,260:DR
AW 320,280:DRAW 320,340:DRAW 340,360
830 PLOT 440,320:DRAW 470,360:DRAW 500,320 [6392]
:DRAW 530,360:DRAW 550,330:DRAW 500,260:DR
AW 470,300:DRAW 440,260:DRAW 440,320
840 MOVE 120,310:FILL 3:MOVE 220,310:FILL  [4795]
3:MOVE 360,310:FILL 2:MOVE 480,310:FILL 3
850 '                                       [117]
860 PLOT 340,320:DRAW 400,320:DRAW 400,300 [4722]
:DRAW 340,300:DRAW 340,320
870 PLOT 330,310:DRAW 340,310:PLOT 400,310 [2402]
:DRAW 410,310
880 PLOT 350,315:DRAW 350,350:PLOT 360,315 [10611]
:DRAW 370,330:DRAW 370,350:PLOT 370,315:DR
AW 380,330:DRAW 380,350:PLOT 380,315:DRAW
390,330:DRAW 390,350:PLOT 392,315:DRAW 392
,325
890 PLOT 350,304:DRAW 340,290:DRAW 340,270 [10582]
:PLOT 360,304:DRAW 350,290:DRAW 350,270:PL
OT 370,304:DRAW 360,290:DRAW 360,270:PLOT
380,304:DRAW 380,270:PLOT 390,304:DRAW 400
,290:DRAW 400,270
900 '                                       [117]
910 PEN 1                                   [549]
920 LOCATE 12,11:PRINT"by GREGOR BILECK"    [2565]
930 LOCATE 1,15:PRINT CHR$(24);"Hiscore of [2958]
 the";CHR$(24)
940 LOCATE 1,16:PRINT CHR$(24);"USER-PROGR [2742]
AM";CHR$(24)
950 LOCATE 1,18:PRINT hiscl                 [1353]
960 PEN 2                                   [548]
970 LOCATE 27,15:PRINTchr$(24)"Hiscore of  [3487]
the";CHR$(24)
980 LOCATE 29,16:PRINT CHR$(24)"MCP.-PROGR [3434]
AM";CHR$(24)
990 LOCATE 34,18:PRINT hisc2                [1295]
1000 '                                      [117]
1010 '                                      [117]
1020 '                                      [117]
1030 INK 1,21:INK 2,16:INK 3,14             [545]
1040 '                                      [117]
1050 IF INKEY(47)=0 OR INKEY(76)=0 THEN 10 [2151]
70 ELSE 1050
1060 '                                      [117]
1070 '---------                             [555]
1080 '                                      [117]
1090 FOR oft=0 TO 15                        [739]
1100 LOCATE 1,1:PRINT CHR$(11)              [832]
1110 NEXT                                   [350]
1120 '                                      [117]
1130 FOR loop=1 TO 5                        [527]
1140 LOCATE 1,25                            [731]
1150 FOR oft=0 TO 18                        [748]
1160 PRINT                                  [361]
1170 NEXT                                   [350]
1180 FOR oft=0 TO 17                        [737]
1190 LOCATE 1,1:PRINT CHR$(11)              [832]
1200 NEXT                                   [350]
1210 NEXT                                   [350]
1220 '                                      [117]
1230 LOCATE 1,25                            [731]
1240 FOR oft=1 TO 26                        [708]
1250 PRINT                                  [361]
1260 NEXT                                   [350]
1270 '                                      [117]
1280 '-------------------------------       [1495]
1290 '                                      [117]
1300 '*** Init                              [483]
1310 '                                      [117]
1320 scl=0:sc2=0                            [743]
1330 level=1                                [567]
1340 '                                      [117]
1350 INK 1,0:INK 2,0:INK 3,0                [672]
```

Listing TRON

```
1360 '                                      [117]
1370 '                                      [117]
1380 '*** Aufbau(Feld)                      [1338]
1390 '                                      [117]
1400 PEN 3                                  [547]
1410 FOR h=1 TO 21:LOCATE 40,h:PRINT w$:NE [2664]
XT
1420 FOR r=35 TO 39:LOCATE r,8:PRINT w$:NE [1402]
XT
1430 FOR r=35 TO 39:LOCATE r,15:PRINT w$:N [1880]
EXT
1440 FOR r=35 TO 39:LOCATE r,21:PRINT w$:N [2020]
EXT
1450 FOR r=34 TO 39:LOCATE r,1:PRINT w$:NE [1686]
XT
1460 '                                      [117]
1470 GOSUB 5120                             [891]
1480 '                                      [117]
1490 PEN 1                                  [549]
1500 LOCATE 35,2:PRINT"User:"               [1900]
1510 LOCATE 35,4:PRINT USING"#####";scl     [2232]
1520 LOCATE 35,5:PRINT"====="               [1854]
1530 LOCATE 35,6:PRINT USING"#####";hiscl   [1279]
1540 PEN 2                                  [548]
1550 LOCATE 35,9:PRINT"Mcp.:"               [2022]
1560 LOCATE 35,11:PRINT USING"#####";sc2    [2304]
1570 LOCATE 35,12:PRINT"====="              [860]
1580 LOCATE 35,13:PRINT USING"#####";hisc2  [2092]
1590 PEN 3                                  [547]
1600 LOCATE 35,16:PRINT"Time"               [1640]
1610 LOCATE 35,17:PRINT"Bonus"              [2053]
1620 '                                      [117]
1630 INK 1,21:INK 2,16:INK 3,14             [545]
1640 '                                      [117]
1650 '                                      [117]
1660 '                                      [117]
1670 '                                      [117]
1680 '--------------------------------      [1570]
1690 'Hauptprogramm                         [1580]
1700 '--------------------------------      [1570]
1710 '                                      [117]
1720 '*** init2                             [469]
1730 '                                      [117]
1740 bonus=300                              [815]
1750 mov1=1:mov2=2                          [1052]
1760 tot=0                                  [411]
1770 '                                      [117]
1780 '                                      [117]
1790 '*** Level? Bonus?                     [2127]
1800 '                                      [117]
1810 GOSUB 4810                             [960]
1820 PEN 1                                  [549]
1830 LOCATE 6,5:PRINT"USER"                 [1746]
1840 LOCATE 6,7:PRINT"Program"              [3080]
1850 PEN 2                                  [548]
1860 LOCATE 25,5:PRINT"MCP."                [1008]
1870 LOCATE 22,7:PRINT"Program"             [1359]
1880 PEN 3                                  [547]
1890 LOCATE 16,6:PRINT"and"                 [2135]
1900 LOCATE 7,11:PRINT"You are now enterin [1814]
g"
1910 LOCATE 12,13:PRINT"LEVEL: ";level      [1210]
1920 LOCATE 9,18:PRINT"LEVEL BONUS is :"    [1601]
1930 LOCATE 10,20:PRINT bonus*level;" POIN [1945]
TS"
1940 LOCATE 37,19:PRINT USING"###";bonus    [3246]
1950 LOCATE 35,20:PRINT"*";level            [1658]
1960 FOR wart=1 TO 4000:NEXT                [2116]
1970 GOSUB 4980                             [935]
1980 '                                      [117]
1990 '                                      [117]
2000 '*** Wegweiser zu den Levels           [2006]
2010 '                                      [117]
2020 GRAPHICS PEN 2:PEN 3                   [842]
2030 ON level GOSUB 3190,3350,3430,3540,36 [3795]
70,3810,3960,4140,4260,4400
2040 '                                      [117]
2050 '                                      [117]
2060 '*** Setzen(d.Schiffe)                 [1180]
2070 '                                      [117]
2080 PEN 3                                  [547]
2090 shx1=RND*30+2:shy1=RND*21+2            [1261]
2100 LOCATE shx1,shy1                       [1780]
2110 IF COPYCHR$(#0)=w$ THEN 2090 ELSE PEN [3355]
 1:LOCATE shx1,shy1:PRINT shu$
2120 '                                      [117]
2130 PEN 3                                  [547]
2140 shx2=RND*30+2:shy2=RND*21+2            [1472]
```

Listing TRON

```
2150 LOCATE shx2,shy2                        [1610]
2160 IF COPYCHR$(#0)=w$ THEN 2140 ELSE PEN  [3016]
 2:LOCATE shx2,shy2:PRINT shd$
2170 '                                       [117]
2180 FOR oft=1 TO 6                          [621]
2190 SOUND 1,60,10,15,4                      [1418]
2200 SOUND 1,0,50,0                          [1400]
2210 NEXT                                    [350]
2220 FOR wart=1 TO 1500:NEXT                 [2568]
2230 '                                       [117]
2240 '                                       [117]
2250 '================================       [2058]
2260 '                                       [117]
2270 '                                       [117]
2280 '*** Go!!! (Das Rennen)                 [1623]
2290 '                                       [117]
2300 IF JOY(0)=1 THEN IF mov1<>2 THEN mov1  [2552]
=1 ELSE mov1=2
2310 IF JOY(0)=2 THEN IF mov1<>1 THEN mov1  [3848]
=2 ELSE mov1=1
2320 IF JOY(0)=4 THEN IF mov1<>4 THEN mov1  [2564]
=3 ELSE mov1=4
2330 IF JOY(0)=8 THEN IF mov1<>3 THEN mov1  [2197]
=4 ELSE mov1=3
2340 '                                       [117]
2350 IF INKEY(0)=0 THEN IF mov2<>2 THEN mo  [2560]
v2=1 ELSE mov2=2
2360 IF INKEY(2)=0 THEN IF mov2<>1 THEN mo  [1928]
v2=2 ELSE mov2=1
2370 IF INKEY(8)=0 THEN IF mov2<>4 THEN mo  [3710]
v2=3 ELSE mov2=4
2380 IF INKEY(1)=0 THEN IF mov2<>3 THEN mo  [2965]
v2=4 ELSE mov2=3
2390 '                                       [117]
2400 '                                       [117]
2410 '                                       [117]
2420 IF mov1<>1 THEN 2440 ELSE shy1=shy1-1  [9307]
:PEN 3:LOCATE shx1,shy1:IF COPYCHR$(#0)=w$
 THEN tot=1:GOTO 2650 ELSE PEN 1:LOCATE sh
x1,shy1:PRINT shu$:PEN 3:LOCATE shx1,shy1+
1:PRINT w$
2430 '                                       [117]
2440 IF mov1<>2 THEN 2460 ELSE shy1=shy1+1  [8061]
:PEN 3:LOCATE shx1,shy1:IF COPYCHR$(#0)=w$
 THEN tot=1:GOTO 2650 ELSE PEN 1:LOCATE sh
x1,shy1:PRINT shd$:PEN 3:LOCATE shx1,shy1-
1:PRINT w$
2450 '                                       [117]
2460 IF mov1<>3 THEN 2480 ELSE shx1=shx1-1  [9774]
:PEN 3:LOCATE shx1,shy1:IF COPYCHR$(#0)=w$
 THEN tot=1:GOTO 2650 ELSE PEN 1:LOCATE sh
x1,shy1:PRINT shl$:PEN 3:LOCATE shx1+1,shy
1:PRINT w$
2470 '                                       [117]
2480 IF mov1<>4 THEN 2500 ELSE shx1=shx1+1  [10904]
:PEN 3:LOCATE shx1,shy1:IF COPYCHR$(#0)=w$
 THEN tot=1:GOTO 2650 ELSE PEN 1:LOCATE sh
x1,shy1:PRINT shr$:PEN 3:LOCATE shx1-1,shy
1:PRINT w$
2490 '                                       [117]
2500 IF mov2<>1 THEN 2520 ELSE shy2=shy2-1  [9093]
:PEN 3:LOCATE shx2,shy2:IF COPYCHR$(#0)=w$
 THEN tot=2:GOTO 2650 ELSE PEN 2:LOCATE sh
x2,shy2:PRINT shu$:PEN 3:LOCATE shx2,shy2+
1:PRINT w$
2510 '                                       [117]
2520 IF mov2<>2 THEN 2540 ELSE shy2=shy2+1  [9271]
:PEN 3:LOCATE shx2,shy2:IF COPYCHR$(#0)=w$
 THEN tot=2:GOTO 2650 ELSE PEN 2:LOCATE sh
x2,shy2:PRINT shd$:PEN 3:LOCATE shx2,shy2-
1:PRINT w$
2530 '                                       [117]
2540 IF mov2<>3 THEN 2560 ELSE shx2=shx2-1  [12014]
:PEN 3:LOCATE shx2,shy2:IF COPYCHR$(#0)=w$
 THEN tot=2:GOTO 2650 ELSE PEN 2:LOCATE sh
x2,shy2:PRINT shl$:PEN 3:LOCATE shx2+1,shy
2:PRINT w$
2550 '                                       [117]
2560 IF mov2<>4 THEN 2590 ELSE shx2=shx2+1  [11456]
:PEN 3:LOCATE shx2,shy2:IF COPYCHR$(#0)=w$
 THEN tot=2:GOTO 2650 ELSE PEN 2:LOCATE sh
x2,shy2:PRINT shr$:PEN 3:LOCATE shx2-1,shy
2:PRINT w$
2570 '                                       [117]
2580 '                                       [117]
2590 PEN 3                                   [547]
2600 bonus=bonus-1                           [881]
2610 LOCATE 35,19:PRINT USING"#####";bonus  [1213]
2620 '                                       [117]
```

Listing TRON

```
2630 GOTO 2290                               [345]
2640 '                                       [117]
2650 '--------------------------------       [1611]
2660 '                                       [117]
2670 '*** Kollision                          [567]
2680 '                                       [117]
2690 SOUND 7+128,0,400,15,2,,15              [2414]
2700 '                                       [117]
2710 INK 0,26                                [64]
2720 FOR wart=1 TO 50:NEXT                   [1724]
2730 INK 0,0                                 [53]
2740 '                                       [117]
2750 IF tot=1 THEN xko=16*shx1:yko=(25-shy  [3817]
1)*16 ELSE xko=16*shx2:yko=(25-shy2)*16
2760 '                                       [117]
2770 FOR far=1 TO 4                          [817]
2780 GRAPHICS PEN far                        [1342]
2790 FOR gr=1 TO 30 STEP 2                   [757]
2800 DEG                                     [133]
2810 MOVE xko,yko+gr                         [1348]
2820 FOR a=0 TO 360 STEP 90                  [1498]
2830 DRAW xko+gr*COS(a),yko+gr*SIN(a)        [1931]
2840 NEXT                                    [350]
2850 NEXT                                    [350]
2860 NEXT                                    [350]
2870 '                                       [117]
2880 '                                       [117]
2890 IF tot=1 THEN GRAPHICS PEN 2 ELSE GRA  [1846]
PHICS PEN 1
2900 FOR y=272 TO 160 STEP-2                 [1015]
2910 SOUND 1,956,1,15,,,1                    [1637]
2920 PLOT 190,y:DRAW 338,y                   [1378]
2930 NEXT                                    [350]
2940 '                                       [117]
2950 IF tot=1 THEN GRAPHICS PEN 1 ELSE GRA  [1852]
PHICS PEN 2
2960 PLOT 188,158:DRAW 188,274:DRAW 340,27  [3334]
4:DRAW 340,158:DRAW 188,158
2970 PLOT 192,162:DRAW 192,270:DRAW 336,27  [3324]
0:DRAW 336,162:DRAW 192,162
2980 '                                       [117]
2990 IF tot=1 THEN PEN 2:LOCATE 15,10:PRIN  [3382]
T"MCP.:" ELSE PEN 1:LOCATE 15,10:PRINT"USE
R:"
3000 PEN 3                                   [547]
3010 LOCATE 15,12:PRINT"Bonus"               [1544]
3020 LOCATE 16,14:PRINT USING"####";bonus   [1598]
3030 '                                       [117]
3040 FOR wart=1 TO 800:NEXT                  [2043]
3050 '                                       [117]
3060 SOUND 1,1911,750,15,3,2:SOUND 2,956,7  [4655]
50,15,3,2:SOUND 4,478,750,15,3,2
3070 '                                       [117]
3080 '                                       [117]
3090 FOR count=(bonus-1) TO 0 STEP-1         [2551]
3100 IF tot=1 THEN PEN 2:sc2=sc2+level:LOC  [7032]
ATE 35,11:PRINT USING"#####";sc2 ELSE PEN
1:sc1=sc1+level:LOCATE 35,4:PRINT USING"##
###";sc1
3110 PEN 3                                   [547]
3120 LOCATE 16,14:PRINT USING"####";count   [1595]
3130 NEXT                                    [350]
3140 '                                       [117]
3150 '                                       [117]
3160 level=level+1                           [1036]
3170 GOSUB 5120                              [891]
3180 IF level=11 THEN GOTO 4590 ELSE GOTO   [3064]
1740
3190 '                                       [117]
3200 '                                       [117]
3210 '--------------------------------       [1570]
3220 '                                       [117]
3230 '*** Levels(Hintergrundmuster)          [3224]
3240 '                                       [117]
3250 '---> 1                                 [516]
3260 '                                       [117]
3270 FOR x=20 TO 508 STEP 8                  [1882]
3280 PLOT x,32:DRAW x,382                    [876]
3290 NEXT                                    [350]
3300 FOR y=32 TO 380 STEP 8                  [2100]
3310 PLOT 16,y:DRAW 510,y                    [181]
3320 NEXT                                    [350]
3330 RETURN                                  [555]
3340 '                                       [117]
3350 '---> 2                                 [519]
3360 '                                       [117]
3370 FOR y=382 TO 32 STEP-16                 [1636]
3380 PLOT 16,y:DRAW 510,y-8                  [260]
3390 NEXT                                    [350]
```

Listing TRON

```
3400 FOR h=6 TO 19:LOCATE 17,h:PRINT w$:NE [3098]
XT
3410 RETURN                                 [555]
3420 '                                      [117]
3430 '---> 3                                [522]
3440 '                                      [117]
3450 FOR x=16 TO 508 STEP 16               [1404]
3460 PLOT x,32:DRAW x,382                   [876]
3470 NEXT                                   [350]
3480 FOR r=13 TO 21:LOCATE r,4:PRINT w$:NE [2512]
XT
3490 FOR r=7 TO 27:LOCATE r,8:PRINT w$:NEX [2133]
T
3500 FOR r=7 TO 27:LOCATE r,17:PRINT w$:NE [3390]
XT
3510 FOR r=13 TO 21:LOCATE r,21:PRINT w$:N [2705]
EXT
3520 RETURN                                 [555]
3530 '                                      [117]
3540 '---> 4                                [525]
3550 '                                      [117]
3560 y=382:x1=16:x2=510                    [1646]
3570 FOR oft=1 TO 22                        [696]
3580 PLOT 16,y:DRAW x1,32:PLOT 510,y:DRAW  [1666]
x2,32
3590 y=y-16:x1=x1+16:x2=x2-16              [2924]
3600 NEXT                                   [350]
3610 FOR oft=1 TO 20                        [690]
3620 r=RND*30+2:h=RND*21+2                  [548]
3630 LOCATE r,h:PRINT w$                   [1646]
3640 NEXT                                   [350]
3650 RETURN                                 [555]
3660 '                                      [117]
3670 '---> 5                                [528]
3680 '                                      [117]
3690 x=264:y=210                            [990]
3700 FOR gr=0 TO 164 STEP 8                [1208]
3710 MOVE x,y+gr                           [1022]
3720 DEG                                    [133]
3730 FOR a=0 TO 360 STEP 45                [1523]
3740 DRAW x+gr*SIN(a),y+gr*COS(a)           [576]
3750 NEXT                                   [350]
3760 NEXT                                   [350]
3770 FOR h=6 TO 19:LOCATE 17,h:PRINT w$:NE [3098]
XT
3780 FOR r=6 TO 28:LOCATE r,12:PRINT w$:NE [2198]
XT
3790 RETURN                                 [555]
3800 '                                      [117]
3810 '---> 6                                [531]
3820 '                                      [117]
3830 FOR oft=1 TO 200                       [565]
3840 x=RND*494+16:y=RND*350+32             [2873]
3850 PLOT x,y:PLOT x+2,y                   [1555]
3860 NEXT                                   [350]
3870 FOR r=7 TO 13:LOCATE r,7:PRINT w$:NEX [2376]
T
3880 FOR r=7 TO 13:LOCATE r,18:PRINT w$:NE [2308]
XT
3890 FOR r=21 TO 27:LOCATE r,7:PRINT w$:NE [3033]
XT
3900 FOR r=21 TO 27:LOCATE r,18:PRINT w$:N [2740]
EXT
3910 FOR h=8 TO 17:LOCATE 7,h:PRINT w$:NEX [3597]
T
3920 FOR h=8 TO 17:LOCATE 27,h:PRINT w$:NE [4163]
XT
3930 FOR r=13 TO 21:LOCATE r,12:PRINT w$:N [2807]
EXT
3940 RETURN                                 [555]
3950 '                                      [117]
3960 '---> 7                                [534]
3970 '                                      [117]
3980 FOR oft=1 TO 20                        [690]
3990 x=RND*494+16:y=RND*350+32             [2873]
4000 PLOT 16,382:DRAW x,y:PLOT 510,32:DRAW [2903]
 x,y
4010 NEXT                                   [350]
4020 FOR r=8 TO 12:LOCATE r,6:PRINT w$:NEX [1679]
T
4030 FOR r=22 TO 26:LOCATE r,6:PRINT w$:NE [2558]
XT
4040 FOR r=22 TO 26:LOCATE r,19:PRINT w$:N [2374]
EXT
4050 FOR r=8 TO 12:LOCATE r,19:PRINT w$:NE [1995]
XT
```

Listing TRON

```
4060 FOR h=7 TO 10:LOCATE 12,h:PRINT w$:NE [2104]
XT
4070 FOR h=15 TO 18:LOCATE 12,h:PRINT w$:N [1638]
EXT
4080 FOR h=7 TO 10:LOCATE 22,h:PRINT w$:NE [1767]
XT
4090 FOR h=15 TO 18:LOCATE 22,h:PRINT w$:N [1666]
EXT
4100 FOR r=12 TO 22:LOCATE r,11:PRINT w$:N [3766]
EXT
4110 FOR r=12 TO 22:LOCATE r,14:PRINT w$:N [3235]
EXT
4120 RETURN                                 [555]
4130 '                                      [117]
4140 '---> 8                                [505]
4150 '                                      [117]
4160 FOR oft=1 TO 100                       [729]
4170 x=RND*478+16:y=RND*334+32             [2137]
4180 PLOT x,y:DRAW x+8,y+16:DRAW x+16,y:DR [4939]
AW x,y
4190 NEXT                                   [350]
4200 h=11:FOR r=7 TO 14:LOCATE r,h:PRINT w [3468]
$:h=h-1:NEXT
4210 h=14:FOR r=7 TO 14:LOCATE r,h:PRINT w [3337]
$:h=h+1:NEXT
4220 h=4:FOR r=20 TO 27:LOCATE r,h:PRINT w [3793]
$:h=h+1:NEXT
4230 h=21:FOR r=20 TO 27:LOCATE r,h:PRINT  [2469]
w$:h=h-1:NEXT
4240 RETURN                                 [555]
4250 '                                      [117]
4260 '---> 9                                [508]
4270 '                                      [117]
4280 x=16:y=32                              [965]
4290 FOR oft=1 TO 31                        [691]
4300 PLOT x,382:DRAW x,y:DRAW 510,y         [878]
4310 x=x+16:y=y+10                          [993]
4320 NEXT                                   [350]
4330 FOR r=7 TO 27:LOCATE r,6:PRINT w$:NEX [2361]
T
4340 FOR r=7 TO 27:LOCATE r,12:PRINT w$:NE [2631]
XT
4350 FOR r=7 TO 27:LOCATE r,19:PRINT w$:NE [3544]
XT
4360 FOR h=7 TO 11:LOCATE 7,h:PRINT w$:NEX [2379]
T
4370 FOR h=13 TO 18:LOCATE 27,h:PRINT w$:N [2571]
EXT
4380 RETURN                                 [555]
4390 '                                      [117]
4400 '---> 10                               [664]
4410 '                                      [117]
4420 x1=260:x2=260:st=2                    [1920]
4430 FOR oft=1 TO 16                        [750]
4440 PLOT x1,382:DRAW x1,32:PLOT x2,382:DR [2334]
AW x2,32
4450 x1=x1-st:x2=x2+st:st=st+2             [1060]
4460 NEXT                                   [350]
4470 LOCATE 15,17:PRINT w$:LOCATE 26,17:PR [1822]
INT w$
4480 FOR r=8 TO 13:LOCATE r,7:PRINT w$:NEX [2147]
T
4490 FOR h=8 TO 16:LOCATE 8,h:PRINT w$:NEX [2447]
T
4500 FOR r=8 TO 13:LOCATE r,17:PRINT w$:NE [2086]
XT
4510 FOR h=12 TO 16:LOCATE 13,h:PRINT w$:N [3047]
EXT
4520 LOCATE 12,12:PRINT w$                 [1217]
4530 FOR h=7 TO 17:LOCATE 18,h:PRINT w$:NE [2441]
XT
4540 FOR r=19 TO 23:LOCATE r,7:PRINT w$:NE  [981]
XT
4550 FOR r=19 TO 23:LOCATE r,12:PRINT w$:N [1585]
EXT
4560 FOR r=19 TO 23:LOCATE r,17:PRINT w$:N [1807]
EXT
4570 FOR h=8 TO 16:LOCATE 24,h:PRINT w$:NE [2306]
XT
4580 RETURN                                 [555]
4590 '                                      [117]
4600 '-------------------------------      [1570]
4610 '                                      [117]
4620 '*** The End                           [616]
4630 '                                      [117]
4640 GOSUB 4810                             [960]
```

Listing TRON

```
4650 PEN 3                                  [547]
4660 LOCATE 10,5:PRINT"G A M E  O V E R"    [2422]
4670 LOCATE 10,9:PRINT"Final Standings:"    [2434]
4680 IF sc1>sc2 THEN PEN 1:LOCATE 10,11:PR [9195]
INT"1.USER : ";sc1:PEN 2:LOCATE 10,13:PRIN
T"2.MCP. : ";sc2 ELSE PEN 2:LOCATE 10,11:P
RINT"1.MCP. : ";sc2:PEN 1:LOCATE 10,13:PRI
NT"2.USER : ";sc1
4690 PEN 3                                  [547]
4700 LOCATE 12,16:PRINT"NICE GAME !"        [1848]
4710 LOCATE 3,18:PRINT"MCP.MAIN PROGRAM is [2333]
 going to"
4720 LOCATE 3,19:PRINT"take the     -Progr [2563]
am to the"
4730 LOCATE 3,20:PRINT"ERASING-CENTER."     [3097]
4740 LOCATE 3,21:PRINT"It is absolutely us [4630]
eless ..."
4750 IF sc1<sc2 THEN PEN 1:LOCATE 12,19:PR [5000]
INT"USER" ELSE PEN 2:LOCATE 12,19:PRINT"MC
P."
4760 '                                      [117]
4770 IF sc1>hisc1 THEN hisc1=sc1            [1832]
4780 IF sc2>hisc2 THEN hisc2=sc2            [2089]
4790 '                                      [117]
4800 IF INKEY(47)=0 OR INKEY(76)=0 THEN GO [4169]
SUB 4980:GOTO 480 ELSE 4800
4810 '                                      [117]
4820 '                                      [117]
4830 '--------------------------------      [1570]
4840 '                                      [117]
4850 '*** Info-Wand auf                     [673]
4860 '                                      [117]
4870 SOUND 1,956,280,15,,1:SOUND 2,1012,28 [2178]
0,15,,1
4880 SOUND 1,0,70,15,1,,1                   [1685]
4890 '                                      [117]
4900 GRAPHICS PEN 2                         [948]
4910 d=382                                  [451]
4920 FOR up=32 TO 208 STEP 2                [1297]
```

Listing TRON

```
4930 PLOT 16,up:DRAW 510,up                 [1074]
4940 PLOT 16,d:DRAW 510,d                   [398]
4950 d=d-2                                  [748]
4960 NEXT                                   [350]
4970 RETURN                                 [555]
4980 '                                      [117]
4990 '*** Info-Wand ab                      [762]
5000 '                                      [117]
5010 SOUND 1,956,280,15,,1:SOUND 2,1012,28 [2178]
0,15,,1
5020 SOUND 1,0,70,15,1,,1                   [1685]
5030 '                                      [117]
5040 GRAPHICS PEN 0                         [950]
5050 d=206                                  [459]
5060 FOR up=208 TO 382 STEP 2               [1537]
5070 PLOT 16,up:DRAW 510,up                 [1074]
5080 PLOT 16,d:DRAW 510,d                   [398]
5090 d=d-2                                  [748]
5100 NEXT                                   [350]
5110 RETURN                                 [555]
5120 '                                      [117]
5130 '                                      [117]
5140 '------------                          [797]
5150 '                                      [117]
5160 '*** Feldrahmen                        [1153]
5170 '                                      [117]
5180 PEN 3                                  [547]
5190 FOR h=1 TO 23:LOCATE 1,h:PRINT w$:NEX [2879]
T
5200 FOR h=1 TO 23:LOCATE 33,h:PRINT w$:NE [2383]
XT
5210 FOR r=1 TO 33:LOCATE r,1:PRINT w$:NEX [2195]
T
5220 FOR r=1 TO 33:LOCATE r,24:PRINT w$:NE [2392]
XT
5230 FOR h=1 TO 24:LOCATE 34,h:PRINT w$:NE [2066]
XT
5240 RETURN                                 [555]
```

Listing TRON

Unter den Sternen

Kleines Geschicklichkeitsspiel für starke Nerven

Ein fesselndes Spiel muß nicht immer mit einem langen programmierten Listing zusammenhängen. Daß es auch anders geht, beweist uns "Starbreaker".

Ähnlichkeiten mit dem Spiel Arkanoid oder Bustout (ein Spiel aus der PC International, Ausgabe 12/87) sind vorhanden. Sie besitzen fünf Leben, von denen Sie eins verlieren, wenn der heruntergefallene Stern nicht durch eine Abwehrbasis zurückkatapultiert wird. Dabei sollte der Stern so abgewehrt werden, daß möglichst viele Steine getroffen werden, die sich am "Himmel" tummeln.

Steuerung

Die Bewegungen der Abwehrbasis werden durch den Joystick (links/rechts und Feuerknopf) ausgeführt.

(Gottfried Stampel/cd)

für 464-664-6128

```
10 'STARBREAKER                             [423]
20 MODE 1:BORDER 3                          [1437]
30 PEN 2:LOCATE 11,24:PRINT"FIRE TO PLAY"   [1407]

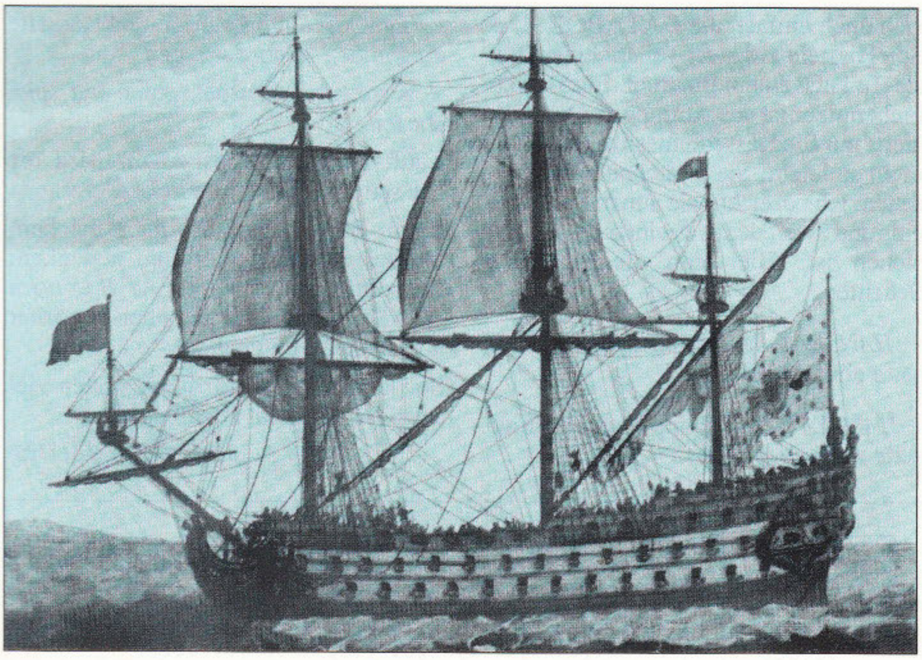
40 IF JOY(0)<>16 THEN 40                    [1237]
50 CLS                                      [91]
60 FOR z=5 TO 36 STEP 3:FOR y=1 TO 8 STEP  [4562]
2:LOCATE z,y:PRINT CHR$(143):NEXT:NEXT
70 b=9:c=5:d=0                              [870]
80 GOSUB 300                                [1016]
90 f=1:i=1:IF INT(RND*2)=1 THEN f=-f        [1719]
100 GOSUB 310                               [1022]
110 ORIGIN 0,400                            [520]
120 x=b+4:y=11:g=x:j=y                      [1324]
130 g=x+f:j=y+i                             [670]
140 IF g=3 OR g=38 THEN f=-f                [1277]
150 GOSUB 280                               [925]
```

Listing Starbreaker

```
160 IF j=24 AND g>b+1 AND g<b+6 THEN i=-i: [4902]
j=j-2:a=((x>b+5)OR(x<b+2)):IF a=-1 THEN f=
f*a:g=g+f:j=j+1
170 IF j=25 THEN LOCATE x,y:PRINT" ":GOTO  [1507]
260
180 GOSUB 310                               [1022]
190 e=TEST((16*g)-1,-(16*j)-1)              [1857]
200 IF e<>0 THEN i=-i:h=g:k=j:j=j+i:LOCATE [1146]
 h,k:PRINT " "
210 IF e=2 THEN d=d+10:GOSUB 300            [1082]
220 IF d=440 THEN 50 ELSE 230               [1700]
230 IF j=1 THEN i=1                         [913]
240 LOCATE  x,y:PRINT " ":LOCATE g,j:PRINT [2321]
 "*":x=g:y=j
250 GOTO 130                                [371]
260 c=c-1:IF c=0 THEN 30                    [1230]
270 GOSUB 300:GOTO 90                       [2018]
280 IF INKEY(74)=0 AND b>2 THEN b=b-2 ELSE [3018]
 IF INKEY(75)=0 AND b<32 THEN b=b+2:RETURN

290 RETURN                                  [555]
300 LOCATE 4,25:PRINT"SCORE:";d:LOCATE 30, [4205]
25:PRINT"LIVES:";c:RETURN
310 LOCATE b,24:PRINT"  ";"----";"  ":RETU [1576]
RN
```

Listing Starbreaker



Golddukaten und Schiffskanonen

Hanseat – ein Abenteuer im Mittelalter

Heutzutage mal schnell geschäftlich nach London, Paris, Hongkong oder San Francisco zu müssen, ist nur noch für diejenigen unmöglich, die das Fliegen nicht vertragen. 'Jetten' sagt der eingefleischte Devisenjäger dazu. Wie müssen sich allerdings die Kaufleute des ausgehenden vierzehnten Jahrhunderts gefühlt haben? Wenn Sie in diese Zeit mal reinschnuppern wollen, bitte sehr: Hier ist Hanseat, die Handelssimulation.

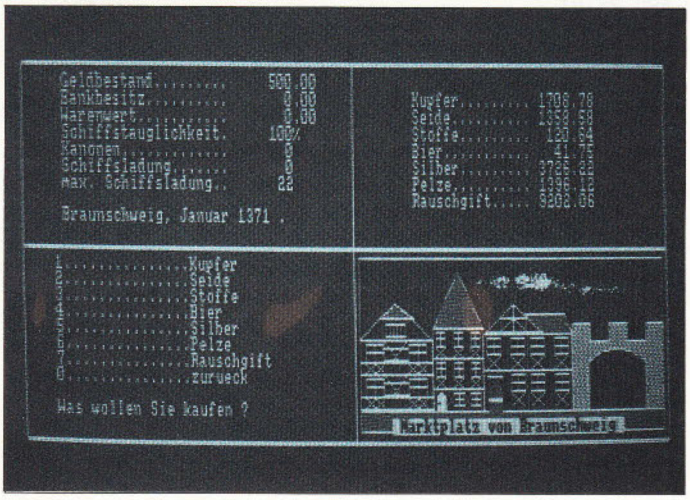
Willkommen bei Hanseat, dem Simulationsspiel in BASIC, mit besonders naturgetreuer Grafikunterstützung. Sie sind ein Händler, der 1371 seine Karriere beginnt und versuchen muß, durch geschicktes An- und Verkaufen besonders viel Geld zu machen, um in die höheren Bevölkerungsschichten vorzudringen. Doch das ist gar nicht so einfach: da gibt es die bösen Piraten, die sie ständig angreifen. Aber mit geschickten Fingern auf der *SPACE-Taste* können Sie sie abwehren. Wenn Sie es partout nicht lassen können, Rauschgift zu schmuggeln, können Sie dies auch tun, allerdings ist das Auge des Gesetzes sehr wachsam.

Das Spiel

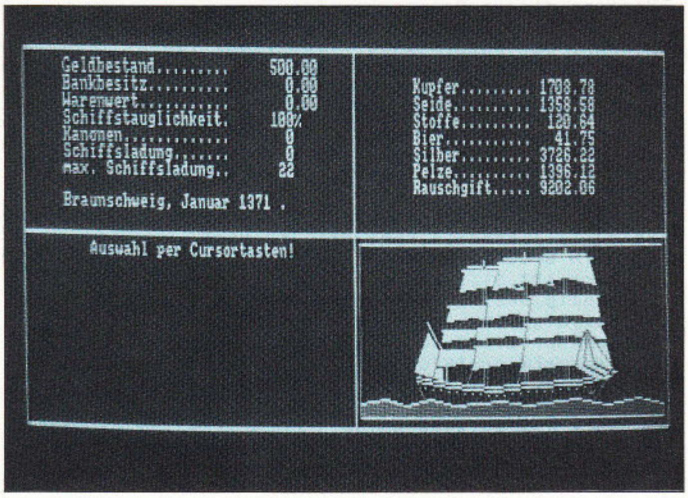
Haben Sie das Spiel abgetippt und mit

```
RUN"HANSEAT.BAS"
```

gestartet, sehen Sie einen kurzen Text, der Sie in das Spiel einführt. Danach geben Sie bitte Ihren Namen ein. Links unten befindet sich nun das Dialogfenster, unten rechts das Grafikfenster, oben rechts das Fenster mit den Preisen und oben links das Statusfenster. Mit der Taste *"K"* können Sie nun Waren kaufen. Dazu brauchen Sie allerdings das nötige Kleingeld. Wieviel Sie brauchen, erfahren Sie im Preisfenster. Es kann allerdings passieren, daß der Händler Ihnen schlechte Waren liefert. Mit der Taste *"V"* verkaufen Sie Waren, sofern Sie welche haben. Drücken Sie die Taste *"W"*, so können Sie in eine andere Stadt reisen, indem Sie mit den Cursortasten AUF und AB die jeweilige Stadt anwählen und dann die ENTER-Taste drücken. Durch das ständige Reisen wird die Qualität Ihres Schiffes auch nicht besser, und somit nimmt auch nach jeder Fahrt die Tauglichkeit ab. Der Abzug richtet sich nach der Anzahl der geladenen Waren und der Entfernung. Wenn Ihr Schiff völlig unbrauchbar geworden ist, wird das Spiel automatisch beendet. Doch Sie brauchen keine Angst zu haben, denn Sie können zwischendurch immer einmal Ihr Schiff reparieren lassen oder ein neues, größeres kaufen. Sie können aber auch Ihr Geld auf der Bank gut anlegen. Das bringt nämlich Zinsen. Bitte drücken Sie im Hauptmenü die Taste *"B"*. Mit der Taste *"Q"* beenden Sie das Spiel vorzeitig. Während dem Spielen gibt es immer wieder Ereignisse, die positiv oder negativ für Sie sind: manchmal können Sie sich Ihr Schiff reparieren lassen oder ein



Hanseat zeichnet sich durch seine grafische Untermalung der einzelnen Schritte aus. Man kann somit jede Handlung genauestens mitverfolgen



Wir befinden uns gerade auf See und das Ziel ist nicht mehr weit

neues kaufen. Kanonen gibt's auch manchmal zu ersteigern. Und nun noch ein paar Worte zu den Piraten: am Anfang haben diese 100 Prozent Kampfenergie. Drücken Sie die *SPACE-Taste*, wenn der Balken auf 'Piraten - 50%' steht, dann wird diesen 50 Prozent ihrer Energie abgezogen. Entsprechend ist es bei den anderen Balken. Treffen Sie z.B. auf 'Schiff - 50%', werden Ihnen 50 Prozent Schiffstauglichkeit abgezogen!

Das Abtippen

Bevor Sie Hanseat spielen dürfen, müssen Sie es abtippen oder besorgen Sie sich doch einfach die DATABOX! Das Spiel besteht aus zwei Teilen: einem Basic-Teil und einem Binärteil. Für Diskettenbenutzer ist das Abtippen unkompliziert, nur die Kassettenbesitzer haben es nicht so leicht: Sie brauchen zwei Kassetten. Ich bezeichne sie nun mit Kassette eins und zwei, damit es einfacher zu verstehen ist. Hier nun die einzelnen Schritte:

– *HANSEAT.BAS* abtippen und auf Kassette eins abspeichern

– *HANSEAT.DAT* abtippen und auf Kassette zwei abspeichern

– *HANSEAT.DAT* starten und Kassette eins in den Recorder legen

– Kassette eins zurückspulen und Spiel starten

Läuft alles ordnungsgemäß, wird die Kassette zwei nicht mehr benötigt.

Bevor Sie sich ans Abtippen machen, müssen Sie eines allerdings bedenken: Sie sollten weder *HANSEAT.BAS* noch *HANSEAT.DAT* starten, ohne es vorher abgespeichert zu haben!

So, jetzt heißt es nur noch, Ihnen viel Spaß mit Hanseat zu wünschen.

(Markus Felder/jb)

für 464-664-6128

```
10 '****************************           [1143]
20 '*     H  A  N  S  E  A  T     *         [678]
30 '*                             *         [175]
40 '*        Version 2.1          *         [98]
50 '*                             *         [175]
60 '*                             *         [175]
70 '* CPC-Version:Markus Felder   *         [3190]
80 '*                             *         [175]
90 '*  Das Vorbild dieses Pro-    *         [2352]
100 '*   gramms ist in der PC-    *         [1493]
110 '*  Databox 7/87 erschienen   *         [1991]
120 '*    Autor:Hartmut Pfarr     *         [1972]
130 '****************************           [1143]
140 '                                       [117]
150 MODE 2:INK 1,24:PEN 1:INK 0,1:BORDER 1 [2954]
:PAPER 0
160 MEMORY &5FFF:LOAD"!hanseat.spr",&6000  [1616]
170 '                                       [117]
180 GOSUB 2870                              [901]
190 '*** Window's und Umrahmung ***         [2198]
200 MODE 2                                  [513]
210 FOR i=2 TO 24:LOCATE 1,i:PRINT CHR$(14 [5665]
9);:LOCATE 80,i:PRINT CHR$(149);:LOCATE 41
,i:PRINT CHR$(149);:NEXT
220 FOR i=2 TO 79:LOCATE i,1:PRINT CHR$(15 [7015]
4);:LOCATE i,24:PRINT CHR$(154);:LOCATE i,
12:PRINT CHR$(154);:NEXT
230 RESTORE 230:FOR i=1 TO 9:READ sy,x,y:L [8190]
OCATE x,y:PRINT CHR$(sy);:NEXT:DATA 150,1,
1,156,80,1,151,1,12,157,80,12,147,1,24,153
,80,24,158,41,1,159,41,12,155,41,24
240 WINDOW 2,40,13,23:WINDOW#1,2,40,2,11   [3435]
250 WINDOW#2,42,79,2,11:WINDOW#3,42,79,13, [2724]
23
260 '*** Namenseingabe **                   [1332]
270 CLS                                     [91]
280 PRINT"Anmerkung!!!":PRINT:PRINT"Die PC [19822]
-Version dieses Programmes ist  unter dem
 Titel '1371' auf der PC-Data-box 7/87 ers
chienen. Der Autor jenes  Programms ist H
artmut Pfarr!":CALL &BB06:CLS
290 LOCATE 2,3:INPUT"Ihr Name: ",name$     [2163]
300 IF name$="" THEN 290                    [317]
310 name$=LEFT$(name$,8)                    [767]
320 '*** Initialisierung ***                [1272]
330 DIM stadt$(10),ware$(7),warenwert(7),w [2170]
are(7),besitz(7)
340 FOR i=1 TO 10:READ stadt$(i),xstadt(i) [4437]
,ystadt(i):ystadt(i)=ystadt(i)*2:NEXT
350 DATA Braunschweig,445,36,Bremen,420,47 [6874]
,Hamburg,437,51,Luebeck,463,56,Rostock,492
,54,Danzig,546,58,Koenigsberg,579,65,Visby
,543,89,Riga,601,79,Reval,614,92
360 bankspr=&6050:schiffspr=&6D3A:marktspr [3880]
=&7A24:kartespr=&870E
370 FOR i=1 TO 7:READ ware$(i),warenwert(i [1219]
):NEXT
380 DATA Kupfer,2000,Seide,1500,Stoffe,150 [5021]
,Bier,50,Silber,4000,Pelze,1500,Rauschgift
,10000
```

Listing HANSEAT

```
390 geld=500:schiff=100:schiffladung=22:la [7520]
dungswert=0:kanonen=0:stadt=1:bank=0:jahr=
1371:in$=CHR$(24):pip$=CHR$(7)
400 DIM mon$(12):mon=1:FOR i=1 TO 12:READ  [9311]
mon$(i):NEXT:DATA Januar,Februar,Maerz,Apr
il,Mai,Juni,Juli,August,September,Oktober,
November,Dezember
410 GOSUB 1810 '===> Preise definieren     [3080]
420 GOSUB 1710 '===> Checkkarte            [1471]
430 '*** Hauptmenue ***                     [954]
440 CLS                                     [91]
450 GOSUB 3070:a$=stadt$(stadt):LOCATE#3,I [5241]
NT((38-LEN(a$))/2),10:PRINT#3,in$+a$+in$;
460 PRINT:PRINT"           Bitte  waehlen Si [3900]
e:":PRINT
470 PRINT"  K..................Waren kauf [3583]
en"
480 PRINT"  V..................Waren verk [3147]
aufen"
490 PRINT"  B..................Bank besuc [2421]
hen"
500 PRINT"  W..................weiterreis [2083]
en"
510 PRINT"  Q..................aufhoeren" [2485]
520 a$=LOWER$(INKEY$):IF a$="" THEN 520    [1526]
530 IF a$="k" THEN GOSUB 3110:GOTO 590     [1700]
540 IF a$="w" THEN 1490                     [1771]
550 IF a$="v" THEN GOSUB 3110:GOTO 890     [3486]
560 IF a$="b" THEN CLS#3:CALL &6000,329,10 [4298]
3,87,38,bankspr:GOTO 1140
570 IF a$="q" THEN 3180                     [1525]
580 GOTO 520                                [413]
590 '*** Waren einkaufen ***                [1148]
600 CLS                                     [91]
610 PRINT"   1...............Kupfer"       [1650]
620 PRINT"   2...............Seide"        [2309]
630 PRINT"   3...............Stoffe"       [1959]
640 PRINT"   4...............Bier"         [1725]
650 PRINT"   5...............Silber"       [2143]
660 PRINT"   6...............Pelze"        [2452]
670 PRINT"   7...............Rauschgift"   [2008]
680 PRINT"   0...............zurueck"      [1630]
690 PRINT                                   [361]
700 PRINT"   Was wollen Sie kaufen ? "     [2177]
710 a$=INKEY$:IF a$="" THEN 710             [1211]
720 IF a$="0" OR a$=CHR$(13) THEN 430      [1956]
730 IF a$<"0" OR a$>"7" THEN 710           [713]
740 kauf=VAL(a$)                            [1045]
750 IF kauf=0 THEN 430                      [320]
760 IF ware(kauf)>geld THEN PRINT pip$" So [6433]
viel Geld haben Sie gar nicht!";:CALL &BB0
6:GOTO 430
770 CLS                                     [91]
780 leisten=INT(geld/ware(kauf))            [1326]
790 PRINT"Sie koennen sich"leisten;:PRINT" [5818]
Stueck davon";:PRINT"leisten."
800 LOCATE 1,6:INPUT"Wieviel wollen Sie ?  [4378]
",stueck
810 IF stueck>leisten OR stueck<0 THEN PRI [3023]
NT CHR$(11);:GOTO 800
820 IF ladung+stueck>schiffladung THEN PRI [8176]
NT"Sie koennen nur "schiffladung" Stueck a
uf Ihr Schiff laden!";:CALL &BB06:GOTO 800
830 geld=geld-(stueck*ware(kauf))          [1554]
840 besitz(kauf)=besitz(kauf)+stueck       [1614]
850 ladung=ladung+stueck                    [810]
```

Listing HANSEAT

```
860 GOSUB 2700                                  [911]
870 GOSUB 1870                                  [861]
880 GOSUB 1710:GOTO 600                         [1035]
890 '*** Waren verkaufen ***                    [821]
900 CLS                                         [91]
910 IF ladung=0 THEN CLS:LOCATE 7,1:PRINT       [5830]
pip$"Sie haben gar keine Waren!":CALL &BB0
6:GOTO 430
920 FOR i=1 TO 7                                [447]
930 PRINT "    ";i;"........"USING"###";bes     [3824]
itz(i);:PRINT" Stueck    "ware$(i)
940 NEXT:PRINT"     0.....................      [2930]
zuruck"
950 PRINT                                       [361]
960 PRINT"    Was wollen Sie verkaufen ? "      [3133]
970 a$=INKEY$:IF a$="" THEN 970                 [1270]
980 IF a$="0" OR a$=CHR$(13) THEN 430           [1956]
990 IF a$<"1" OR a$>"7" THEN 970                [2103]
1000 verkauf=VAL(a$)                            [945]
1010 IF besitz(verkauf)<1 THEN LOCATE 4,11      [7584]
:PRINT pip$"Sie haben gar keine solche War
e!";:CALL &BB06:GOTO 890
1020 CLS                                        [91]
1030 PRINT"Sie haben"besitz(verkauf)"Stuec      [3926]
k "ware$(verkauf)".":PRINT
1040 LOCATE 1,6                                 [626]
1050 INPUT"Wieviel wollen Sie verkaufen? "      [3375]
,stueck
1060 IF stueck>besitz(verkauf) THEN PRINT       [3269]
CHR$(11);:GOTO 1050
1070 geld=geld+(stueck*ware(verkauf))           [2723]
1080 besitz(verkauf)=besitz(verkauf)-stuec      [1984]
k
1090 ladung=ladung-stueck                       [629]
1100 ladungswert=ladungswert-(stueck*ware(      [2728]
verkauf))
1110 GOSUB 1870                                 [861]
1120 GOSUB 1710                                 [869]
1130 GOTO 430                                   [462]
1140 '*** Bank besuchen ***                     [2366]
1150 CLS                                        [91]
1160 LOCATE 7,3:PRINT"E...........Geld ei       [4233]
nzahlen"
1170 LOCATE 7,4:PRINT"A...........Geld ab       [3167]
heben"
1180 LOCATE 7,5:PRINT"V...........Bank ve       [3102]
rlassen"
1190 LOCATE 12,9:PRINT"Bitte wahlen Sie!"       [2776]
1200 a$=LOWER$(INKEY$):IF a$="" THEN 1200       [1565]
1210 IF a$="e" THEN 1250                        [999]
1220 IF a$="a" THEN 1370                        [832]
1230 IF a$="v" THEN 430                         [1283]
1240 GOTO 1200                                  [363]
1250 '*** Geld einzahlen ***                    [2072]
1260 CLS                                        [91]
1270 IF geld<0 OR geld=0 THEN PRINT"Sie ha      [5359]
ben gar kein Geld!":CALL &BB06:GOTO 1140
1280 PRINT"Sie haben "geld" Gulden."            [2194]
1290 PRINT:PRINT"Die Bank bietet Ihnen ein      [9837]
en Zins von   monatlich 1%.":PRINT:PRINT:I
NPUT"Wieviel wollen Sie einzahlen?",einzah
l
1300 IF einzahl>geld THEN 1260                  [1142]
1310 IF einzahl<0 THEN 1260                     [825]
1320 geld=geld-einzahl                          [1417]
1330 bank=bank+einzahl                          [1329]
1340 GOSUB 1870                                 [861]
1350 GOSUB 1710                                 [869]
1360 GOTO 1140                                  [311]
1370 '*** Geld abheben ***                      [1680]
1380 CLS                                        [91]
1390 IF bank<=0 THEN PRINT pip$"Sie haben       [3991]
gar kein Guthaben auf der    Bank.":CALL &
BB06:GOTO 1140
1400 PRINT"Sie haben"bank"Gulden auf der B      [2388]
ank."
1410 LOCATE 1,6:INPUT"Wieviel wollen Sie a      [4341]
bheben? ",abheb
1420 IF abheb=0 THEN 1140                       [623]
1430 IF abheb>bank OR abheb<0 THEN PRINT C      [885]
HR$(11);:GOTO 1410
1440 geld=geld+abheb                            [1021]
1450 bank=bank-abheb                            [666]
1460 GOSUB 1870                                 [861]
1470 GOSUB 1710                                 [869]
1480 GOTO 1140                                  [311]
1490 CLS                                        [91]
1500 y=stadt                                    [662]
1510 LOCATE 8,1:PRINT"Auswahl per Cursorta      [3215]
sten!"
```

Listing HANSEAT

```
1520 WINDOW SWAP 3,0                            [1176]
1530 a$=INKEY$:IF a$="" THEN 1530               [1455]
1540 yy=y                                       [552]
1550 IF INKEY(2)=0 THEN y=y+1                   [1052]
1560 IF INKEY(0)=0 THEN y=y-1                   [1657]
1570 IF INKEY(18)=0 OR INKEY(6)=0 THEN 164      [1085]
0
1580 IF y>10 THEN y=1 ELSE IF y<1 THEN y=1      [1557]
0
1590 xx=xstadt(yy):yy=ystadt(yy):PLOT xx,y      [4073]
y,0:GOSUB 3090
1600 xx=xstadt(y):yy=ystadt(y):PLOT xx,yy,      [4184]
1:GOSUB 3090
1610 LOCATE 6,10:PRINT SPACE$(20);:a$=" "+      [4957]
stadt$(y)+" ":LOCATE ROUND((38-LEN(a$))/2,
0),10:PRINT in$+a$+in$
1620 SOUND 1,142,10:SOUND 2,119,10:SOUND 4      [2113]
,95,10
1630 GOTO 1530                                  [393]
1640 IF y=stadt THEN y=yy:GOTO 1530             [2009]
1650 p=SQR(ABS(xstadt(y)-xstadt(stadt))^2+      [6413]
ABS(ystadt(y)-ystadt(stadt))^2):p=ROUND(p/
15,0):GOSUB 3140:stadt=y:mon=mon+1
1660 IF mon=13 THEN mon=1:jahr=jahr+1           [2801]
1670 GOSUB 2780                                 [831]
1680 WINDOW SWAP 0,3                            [1029]
1690 IF RND>0.5 THEN GOSUB 2010                 [1234]
1700 GOTO 410                                   [450]
1710 '*** Checkkarte ***                        [1179]
1720 GOSUB 1870                                 [861]
1730 LOCATE#1,1,1:PRINT#1,"      Geldbestand    [4641]
........."USING"########.##";geld
1740 PRINT#1,"     Bankbesitz.........."USI     [3525]
NG"########.##";bank
1750 PRINT#1,"     Warenwert..........."USI     [4802]
NG"########.##";ladungswert
1760 PRINT#1,"     Schiffstauglichkeit."USI     [4251]
NG"########";schiff;:PRINT#1,"%"
1770 PRINT#1,"     Kanonen............";US      [2946]
ING"########";kanonen
1780 PRINT#1,"     Schiffsladung......."USI     [3380]
NG"########";ladung
1790 PRINT#1,"     max. Schiffsladung..";US     [4772]
ING"########";schiffladung
1800 PRINT#1:PRINT#1,CHR$(18)"     "stadt$(     [3871]
stadt)", "mon$(mon);jahr".";:RETURN
1810 '*** Preise definieren ***                 [2072]
1820 LOCATE#2,1,2:FOR i=1 TO 7                  [828]
1830 ware(i)=warenwert(i)+0.2*warenwert(i)      [7331]
*RND-0.2*warenwert(i):ware(i)=ROUND(ware(i
),2)
1840 PRINT#2,"        "ware$(i);STRING$(15-     [3749]
LEN(ware$(i)),".");USING"#####.##";ware(i)
1850 NEXT                                       [350]
1860 RETURN                                     [555]
1870 '*** Werte runden ***                      [1032]
1880 ladungswert=0:FOR i=1 TO 7:ladungswer      [4987]
t=ladungswert+(besitz(i)*ware(i)):NEXT
1890 geld=ROUND(geld,2)                         [1113]
1900 warenwert=ROUND(warenwert,2)               [3012]
1910 bank=ROUND(bank,2)                         [1294]
1920 IF schiff>100 THEN schiff=100              [1210]
1930 IF geld>99999999 THEN geld=99999999        [1527]
1940 ladungswert=ROUND(ladungswert,2)           [2410]
1950 IF geld<0 THEN geld=0                      [910]
1960 IF warenwert<0 THEN warenwert=0            [1589]
1970 IF bank<0 THEN bank=0                      [1690]
1980 IF ladungswert<0 THEN ladungswert=0        [2553]
1990 RETURN                                     [555]
2000 GOTO 2170                                  [361]
2010 '*** Ereignisse ***                        [612]
2020 IF RND>0.8 THEN 2690                       [330]
2030 a=INT(RND*10)+1                            [1364]
2040 IF a=2 THEN 2110                           [473]
2050 IF a=3 THEN 2350                           [875]
2060 IF a=4 THEN 2450                           [467]
2070 IF a=5 THEN 2500                           [1016]
2080 IF a=6 THEN 2590                           [701]
2090 IF a=7 OR a=8 THEN 2350                    [826]
2100 GOTO 2690                                  [393]
2110 CLS                                        [91]
2120 PRINT"    Es greift ein Piratenschiff      [5098]
an."
2130 pirat=100:LOCATE 10,3:PRINT"Piratensc      [2602]
hiff:100%"
2140 RESTORE 2140:FOR ww=1 TO 6:READ p$(ww      [9516]
):LOCATE 13,4+ww:PRINT p$(ww):NEXT:DATA Pi
raten -10%,Schiff  -50%,Piraten -20%,Schif
f  -20%,Piraten -50%,Schiff  -10%
```

Listing HANSEAT

```
2150 FOR ww=1 TO 6                              [498]
2160 IF ww<>1 THEN LOCATE 11,3+ww:PRINT"        [6252]
"p$(ww-1)"  " ELSE LOCATE 11,10:PRINT"   "p
$(6)"  "
2170 LOCATE 12,4+ww:PRINT in$" "p$(ww)" "i     [2459]
n$;
2180 SOUND 1,10*ww+300,2,12:FOR p=0 TO kan     [3555]
onen*10:NEXT:IF INKEY$=" " THEN 2200
2190 CALL &BD19:NEXT:GOTO 2150                  [1007]
2200 'Taste gedrueckt                           [833]
2210 IF ww=1 THEN pirat=pirat-10 ELSE IF w     [10469]
w=2 THEN schiff=schiff-50 ELSE IF ww=3 THE
N pirat=pirat-20 ELSE IF ww=4 THEN schiff=
schiff-20 ELSE IF ww=5 THEN pirat=pirat-50
 ELSE IF ww=6 THEN schiff=schiff-10
2220 IF pirat<=0 THEN pirat=0:LOCATE 24,3:     [4936]
PRINT" 0":CALL &BB06:GOTO 2280
2230 IF schiff<=0 THEN schiff=0:GOSUB 1710     [3339]
:CALL &BB06:GOTO 2320
2240 LOCATE 24,3:PRINT USING"###";pirat        [2282]
2250 LOCATE 11,4+ww:PRINT"   "p$(ww)"  "        [2762]

2260 GOSUB 1710                                 [869]
2270 GOTO 2150                                  [349]
2280 'Belohnung                                 [1597]
2290 CLS:PRINT"Herzlichen Glueckwunsch!"        [2160]
2300 PRINT:PRINT"Sie haben unter Einsatz I     [12183]
hres Lebens  die freien Hansestaedte vor
Piraten ge-schuetzt.":PRINT:PRINT"Dafuer g
ebuehrt Ihnen Lohn und Dank!"
2310 belohn=INT(RND*500)+800:PRINT:PRINT"S     [10626]
ie bekommen eine Belohnung in Hoehe   von"
belohn"Gulden.":geld=geld+belohn:CALL &BB0
6:GOSUB 1710:RETURN
2320 'Schiff kaputt                             [1429]
2330 CLS:PRINT"Tut mir leid!!!":PRIN           [2748]
T
2340 PRINT"Die boesen,boesen Piraten zerst     [15031]
oerten  Ihr Schiff! Damit waere jetzt Ihre
 Karriere beendet!":PRINT:PRINT"Auf Wi
edersehen im naechsten Spiel!":CALL &BB06:
GOTO 3180
2350 '** Schiff reparieren **                  [1097]
2360 CLS                                        [91]
2370 a=INT(RND*geld)+geld\2:PRINT"Ein Zimm     [11043]
ermann bietet Ihnen an, fuer  einen Lohn
von"a;:PRINT"Gulden, Ihr Schiff";:PRINT"zu
 reparieren."
2380 IF geld<a THEN PRINT:PRINT:PRIN           [6994]
T"Ihr Geld reicht aber nicht aus!":CALL &B
B06:GOTO 2690
2390 PRINT:PRINT:PRINT:PRINT"Einverstanden     [3948]
 (j/n) ?"
2400 a$=LOWER$(INKEY$):IF a$="" THEN 2400      [1357]
2410 IF a$="j" THEN 2440                        [1372]
2420 IF a$="n" THEN 2690                        [882]
2430 GOTO 2400                                  [347]
2440 geld=geld-a:schiff=100:GOSUB 1710:GOT     [2360]
O 2690
2450 '** Handelspolizei **                      [745]
2460 CLS:PRINT"Die Handelspolizei geht an      [7486]
Bord.Sie   untersucht alles gruendlich."
2470 IF besitz(7)=0 THEN PRINT:PRINT:PRINT     [11204]
:PRINT"Da Sie kein Rauschgift besitzen, ve
r-  laesst sie das Schiff und wuenscht
 Ihnen eine gute Fahrt.":CALL &BB06:GOTO 2
690
2480 PRINT:PRINT:PRINT:PRINT"Sie entdeckte     [6502]
 bei Ihnen"besitz(7)"Paeckchen":PRINT"Raus
chgift. Sie wurden verhaftet."
2490 CALL &BB06:GOTO 3180                       [1309]
2500 '*** neues Schiff kaufen ***               [1424]
2510 b=INT(RND*40)+schiffladung+1:CLS:PRIN     [13752]
T"Ein alter Schiffsbesitzer bietet Ihnen e
in nagelneues Schiff fuer"b"Waren-":PRINT"
einheiten an.":a=INT(RND*geld)+2000+1:PRIN
T:PRINT"Der Preis betraegt"a"Gulden."
2520 IF a>geld THEN PRINT:PRINT"Leider hab     [5870]
en Sie zuwenig Geld!":CALL &BB06:GOTO 2690
2530 PRINT:PRINT"Nehmen Sie an (j/n) ?"         [1918]
2540 a$=LOWER$(INKEY$):IF a$="" THEN 2540      [1345]
2550 IF a$="j" THEN 2580                        [1440]
2560 IF a$="n" THEN 2690                        [882]
2570 GOTO 2540                                  [367]
2580 geld=geld-a:schiff=100:schiffladung=b     [4480]
:GOSUB 1710:GOTO 2690
2590 '** Kanone **                              [778]
2600 CLS                                        [91]
```

Listing HANSEAT

```
2610 preis=INT(RND*geld)+1500:PRINT"Man bi     [4923]
etet Ihnen eine nagelneue Kanone fuer"prei
s"Gulden an."
2620 IF preis>geld THEN PRINT:PRINT:PRINT"     [5906]
Leider haben Sie zuwenig Geld!":CALL &BB06
:GOTO 2690
2630 PRINT:PRINT:PRINT"Wollen Sie sie habe     [3093]
n (j/n) ?"
2640 a$=LOWER$(INKEY$):IF a$="" THEN 2640      [1565]

2650 IF a$="j" THEN 2680                        [1452]
2660 IF a$="n" THEN 2690                        [882]
2670 GOTO 2640                                  [363]
2680 geld=geld-preis:kanonen=kanonen+1:GOS     [3329]
UB 1710:GOTO 2690
2690 RETURN                                     [555]
2700 '*** Schlechte Ware geliefert ***          [2786]
2710 zz=RND:IF zz<0.95 THEN 2770               [2772]
2720 CLS:PRINT"Der Haendler hat Ihnen schl     [12632]
echte Ware  geliefert und ist in eine ande
re Stadt gezogen. Ihre neue Ladung ist des
halb  nichts wert!";
2730 ladungswert=ladungswert-(ware(kauf)*b     [3812]
esitz(kauf))
2740 besitz(kauf)=besitz(kauf)-stueck          [1963]

2750 ladung=ladung-stueck                      [629]
2760 CALL &BB06                                 [393]
2770 RETURN                                     [555]
2780 '*** Zinsen und Schiffstauglichkeit *     [2373]
**
2790 bank=bank+(bank/100)                       [1452]
2800 m=ROUND((RND*5)*(ladung/schiffladung)     [1975]
,0)
2810 schiff=schiff-m-CINT(p/4):GOSUB 1710      [3098]
2820 IF schiff<=0 THEN 2840                     [1024]
2830 RETURN                                     [555]
2840 WINDOW SWAP 3,0:CLS                        [1571]
2850 PRINT"Ihr Schiff ist total kaputt!"        [2824]
2860 CALL &BB06:GOTO 3180                       [1309]
2870 '*** Eroeffnung ***                        [743]
2880 WINDOW#1,1,80,1,25                         [988]
2890 WINDOW 2,79,2,23                           [982]
2900 FOR i=2 TO 24:LOCATE#1,1,i:PRINT#1,CH     [5121]
R$(149);:LOCATE#1,80,i:PRINT#1,CHR$(149);:
NEXT
2910 FOR i=2 TO 79:LOCATE#1,i,1:PRINT#1,CH     [5149]
R$(154);:LOCATE#1,i,24:PRINT#1,CHR$(154);:
NEXT
2920 RESTORE 2920:FOR i=1 TO 4:READ x,y,sy     [4456]
:LOCATE#1,x,y:PRINT#1,CHR$(sy);:NEXT:DATA
1,1,150,80,1,156,1,24,147,80,24,153
2930 '* Grafik *                                [1229]
2940 MOVE 10,350:DRAWR 10,-50:DRAWR 10,50:     [3733]
DRAWR 10,-50:DRAWR 10,50
2950 MOVE 11,350:DRAWR 10,-50:DRAWR 10,50:     [5379]
DRAWR 10,-50:DRAWR 10,50
2960 LOCATE 35,1:PRINT CHR$(24);" HANSEAT      [6462]
";CHR$(24);:LOCATE 56,1:PRINT"von Markus F
elder ";CHR$(164)" DMV";
2970 WINDOW 8,79,4,8                            [1019]
2980 PRINT"ir  schreiben  das Jahr 1371. E     [19189]
s ist die Bluetezeit der deutschen Hanse.S
ie  haben die Aufgabe, als kleiner Kaufman
n mit einem geringen Anfangs-kapital  viel
 Geld zu verdienen und damit in die hoeher
en Bevoelkerungs-schichten vorzudringen."
2990 WINDOW 2,79,8,23                           [996]
3000 PRINT"Das  ist natuerlich nicht leich     [21793]
t. Sie muessen von Stadt zu Stadt reisen,
Warenkaufen bzw. verkaufen,die Bank besuch
en, gegen Piraten kaempfen und so weiter.A
uch  muessen  Sie  sich  gegen  jegliche
Betruegereien  seitens der Haendler";
3010 PRINT"schuetzen. Es  kommt auf Ihre G     [23152]
eschicklichkeit an, ob Sie die Piraten abw
ehrenkoennen. Haben  Sie  zittrige  Haende
 oder aehnliches, dann geht das eben aufK
osten Ihres Schiffes. Ist  es  kaputt,  da
nn  ist  das  Spiel  eben  beendet.";
3020 PRINT"Allerdings  koennen  Sie  waehr     [24214]
end des  Spiels  oefters Ihr Schiff repar
ierenlassen  und auch ein neues kaufen. La
ssen Sie sich aber nicht von der Handels-p
olizei  erwischen  waehrend Sie Rauschgift
 transportieren. Es waere schade um";
3030 PRINT"Ihre Karriere."                      [2312]
3040 PRINT:PRINT:PRINT"            Ich         [7952]
wuensche Ihnen viel Spass mit diesem Spiel
!"
```

Listing HANSEAT

```
3050 LOCATE 29,16:PRINT"Bitte Taste drueck [3598]
en";
3060 FOR ww=1 TO 100:CALL &BB09:NEXT:CALL  [2403]
&BB06:RETURN
3070 'Karte+Stadt zeichnen                 [1157]
3080 CLS#3:CALL &6000,329,102,87,38,kartes [5316]
pr:xx=xstadt(stadt):yy=ystadt(stadt)
3090 PLOT xx-1,yy+2:DRAWR 4,0:DRAWR 0,-2:D [3577]
RAWR -4,0:DRAWR 0,-2:DRAWR 4,0
3100 RETURN                                [555]
3110 'Markt anzeigen                       [1733]
3120 CLS#3:CALL &6000,329,103,87,38,markts [2358]
pr
3130 a$=" Marktplatz von "+stadt$(stadt)+" [6739]
 ":LOCATE#3,ROUND((38-LEN(a$))/2,0),11:PRI
NT#3,in$+a$+in$;:RETURN
3140 'Schiff darstellen                    [1470]
3150 CLS:CALL &6000,329,103,87,38,schiffsp [1521]
r
3160 FOR t=1 TO ROUND(p/2,0):FOR w=1 TO 15 [8779]
:SOUND 1,0,7,w,,,3:NEXT:FOR w=15 TO 1 STEP
 -1:SOUND 1,0,7,w,,,3:NEXT w,t
3170 RETURN                                [555]
3180 '*** Ende, Bewertung ***              [2165]
3190 CLS                                   [91]
3200 egeld=geld+bank+(ladungswert/5)       [2519]
3210 IF egeld<1000 THEN rang$="kleiner Kau [2051]
fmann":ziel=0:GOTO 3270
3220 IF egeld<10000 THEN rang$="mittlerer  [4064]
Kaufmann":ziel=2:GOTO 3270
3230 IF egeld<30000 THEN rang$="Grosshaend [3157]
ler":ziel=3:GOTO 3270
3240 IF egeld<50000 THEN rang$="Patrizier" [3047]
:ziel=4:GOTO 3270
3250 IF egeld<100000 THEN rang$="Patrizier [6346]
 und kaiserlicher Hoflieferant":ziel=5:GOT
O 3270
3260 IF egeld>100000 THEN rang$="Maechtigs [7966]
ter Mann des Staates nach dem Kaiser":ziel
=6
3270 PRINT name$",":PRINT"Ihr Rang ist: "; [2871]
rang$
3280 PRINT:PRINT:PRINT                     [1082]
3290 IF ziel=0 THEN PRINT"Sie haben Ihr Zi [5288]
el nicht erreicht!"
3300 IF ziel=1 THEN PRINT"Sie haben Ihr Zi [3603]
el kaum erreicht!"
3310 IF ziel=2 THEN PRINT"Sie haben Ihr Zi [5293]
el nur bedingt erreicht!"
3320 IF ziel=3 THEN PRINT"Sie haben Ihr Zi [7446]
el mit Aussnahmen voll erreicht!"
3330 IF ziel=4 THEN PRINT"Sie haben Ihr Zi [4922]
el voll erreicht!"
3340 IF ziel=5 THEN PRINT"Sie haben Ihr Zi [5169]
el viel zu niedrig gestellt!"
3350 IF ziel=6 THEN PRINT"GRATULIERE!!!  S [7484]
ie haben Ihr Ziel hervorragend erreicht!!!
"
3360 score=(((geld+bank+ladungswert)/(kano [5938]
nen+1))):score=ROUND(score,0)
3370 PRINT"PUNKTE:";score                  [1824]
3380 PRINT:PRINT"Neues Spiel (j/n) ?"      [1813]
3390 FOR i=0 TO 35 STEP 5:SOUND 1,70-i,10, [5753]
12:SOUND 2,100-i,10,12:SOUND 4,120-i,10,12
:FOR t=1 TO 10:NEXT t,i
3400 a$=LOWER$(INKEY$):IF a$="" THEN 3400  [1349]
3410 IF a$="j" THEN RUN                    [1227]
3420 IF a$="n" THEN 3440                   [860]
3430 GOTO 3400                             [497]
3440 MODE 2:END                            [1120]
```

```
10 '*************************              [894]
20 '*  H  A  N  S  E  A  T  *              [678]
30 '*                       *              [175]
40 '* erzeugt:  HANSEAT.SPR *              [2714]
50 '*                       *              [175]
60 '* Autor:  Markus Felder *              [955]
70 '*************************              [894]
80 MODE 2:PRINT"Der Datalader beschreibt n [16920]
un den Bildschirm. Bitte Datentraeger einl
egen und Taste druecken !!!":CALL &BB06:CL
S:adr=&C000:FOR zeile=120 TO 8430 STEP 10:
READ b$,check$
90 summe=0:FOR w=0 TO 15:b=VAL("&"+MID$(b$ [7714]
,w*2+1,2)):POKE adr,b:summe=summe+b:adr=ad
r+1:NEXT w
100 IF summe<>VAL("&"+check$) THEN CLS:PRI [6677]
NT CHR$(7)"Datafehler in Zeile"zeile". Bit
te korrigieren und neu starten!!!":END
110 NEXT zeile:SAVE"!hanseat.spr",b,&C000, [5334]
&33F8:CLS:PRINT CHR$(7)"Fertig.":END
120 DATA DD6E00DD6601224760DD7E02324960DD, [2071]
66D
130 DATA 7E04324A60DD6E06DD6607DD5E08DD56, [2629]
66F
140 DATA 09CD1DBCED4B4960ED5B4760C5E51AAE, [2299]
7F1
150 DATA 7723130D20F8E17CF6C0C60867300401, [1704]
64F
160 DATA 50C009C110E6C9000000000000000000, [1765]
399
170 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
180 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
190 DATA FFFFFFFFFFFF80000000000000000000, [3134]
67A
200 DATA 00000000000000000000000000000000, [2181]
000
210 DATA 00000000000000000000000180000000, [1862]
081
220 DATA 00000000000000000000000000000000, [2181]
000
230 DATA 00000000000000000000000000000000, [2181]
000
240 DATA 00018000000000000000000000000000, [1584]
081
250 DATA 00000000000000000000000000000000, [2181]
000
260 DATA 00000000000000018000000000000000, [1710]
081
270 DATA 00000000000000000000000000000000, [2181]
000
280 DATA 00000000000000000000000000018000, [2115]
081
290 DATA 00000000000000000000000000000000, [2181]
000
300 DATA 00000000000000000000000000000000, [2181]
000
310 DATA 00000001800000000000000000000000, [1774]
081
320 DATA 00000000000000000000000000000000, [2181]
000
330 DATA 00000000000000000001800000000000, [1894]
081
340 DATA 00000000000000000000000000000000, [2181]
000
350 DATA 00000000000000000000000000000001, [2168]
001
360 DATA 80000000000000000000000000000000, [1274]
080
370 DATA 00000000000000000000000000000000, [2181]
000
360 DATA 00000000018000000000000000000000, [1870]
081
390 DATA 00000000000000000000000000000000, [2181]
000
400 DATA 00000000000000000000000180000000, [1862]
081
410 DATA 00000000000000000000000000000FF8, [2371]
107
420 DATA 00000000000000000000000000000000, [2181]
000
430 DATA 00018000000000000000000000000000, [1584]
081
440 DATA 0000003FFF7FFE000000000000000000, [1129]
2BB
450 DATA 00000000000000018000000000000000, [1710]
081
460 DATA 00000000000000007FDF8C18FDFF0000, [2154]
3FE
470 DATA 00000000000000000000000000018000, [2115]
081
480 DATA 00000000000000000000000000FF83E0, [1941]
262
490 DATA 300603E0FF8000000000000000000000, [2428]
298
500 DATA 00000001800000000000000000000000, [1774]
081
510 DATA 0003FF007C00C001001F007FE0000000, [2526]
3BD
520 DATA 00000000000000000001800000000000, [1894]
081
530 DATA 00000000000007FC000F80030000C000, [1446]
255
```

Listing HANSEAT

```
540 DATA FC001FF0000000000000000000000001, [1680]
20C
550 DATA 80000000000000000000000FF80003F0, [1834]
27A
560 DATA 000C0000300003E0000FFC0000000000, [3099]
22A
570 DATA 00000000000180000000000000000000, [1870]
081
580 DATA 3FF000007C00001000000C00001F0000, [1811]
1E6
590 DATA 03FE0000000000000000000180000000, [1866]
182
600 DATA 00000000007FC000000F800000600000, [1900]
22E
610 DATA 03000000FC000001FF00000000000000, [2348]
1FF
620 DATA 00018000000000000001FF80000001F0, [1773]
2F2
630 DATA 00000180000000800000003E0000000FF, [1245]
2E3
640 DATA C000000000000018000000000000003FE, [2021]
242
650 DATA 00000003E00000006000000000600000, [1449]
0A4
660 DATA 001F00000003FE000000000000018000, [1662]
1BF
670 DATA 00000007FC000000000FC00000001800, [2242]
1EA
680 DATA 00000018000000000FC000000001FF000, [1842]
223
690 DATA 00000001800000001FF8000000001F0, [1824]
289
700 DATA 00000000600000000000600000000003E0, [1686]
149
710 DATA 000000000FFC000000018000003FFFF, [919]
38D
720 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
730 DATA FFFFFFFFFFFFFFFFFFFFFFFFF0000001, [3006]
CE5
740 DATA 80000003FFFFFFFFFFFFFFFFFFFFFFFF, [2459]
C77
750 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
760 DATA FFFFF000000180000003FFFFFFFFFFF0, [3204]
95D
770 DATA 0003FFFFFFF0001FFFFFFE00FC0FFFFF, [1774]
A14
780 DATA FC01C01FFFFFFFFF0000018000003, [1613]
74C
790 DATA FFFFFFFFFF007007FFFFF803803FFFF, [2950]
B28
800 DATA FE00FC0FFFFFFC0000FFFFFFFFFFF000, [1461]
AEE
810 DATA 000180000003FFFFFFFFFFF007007FFF, [2585]
7F4
820 DATA FF803803FFFFFE001C0FFFFFFC0007FF, [2267]
8E1
830 DATA FFFFFFFFF000000180000003FFFFFFFF, [2738]
96C
840 DATA FFF00003FFFFFF800003FFFFFE00000F, [1952]
87D
850 DATA FFFFFC003FFFFFFFFFFFF0000018000, [2605]
9A5
860 DATA 0003FFFFFFFFFFF007007FFFFF803803, [2304]
92D
870 DATA FFFFFE00E00FFFFFFC0007FFFFFFFFFF, [2959]
BE7
880 DATA F000000180000003FFFFFFFFFFF00700, [1597]
766
890 DATA 7FFFFF803803FFFFFE00FC0FFFFFFC00, [3004]
A39
900 DATA 00FFFFFFFFFFF000000180000003FFFF, [1509]
86D
910 DATA FFFFFFF00003FFFFFF803803FFFFFE00, [2037]
AA4
920 DATA FC0FFFFFFC01C01FFFFFFFFFF0000001, [2777]
9D2
930 DATA 80000003FFFFFFFFFFFFFFFFFFFFFFFF, [2459]
C77
940 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
950 DATA FFFFF000000180000003FFFFFFFFFFFF, [3503]
96C
960 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
970 DATA FFFFFFFFFFFFFFFF000000180000003, [2872]
96C
980 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF, [3602]
FF0
```

Listing HANSEAT

```
990 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFF000, [3086]
EE2
1000 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1010 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1020 DATA AAAABB6DB000001800000036DB75555 [2229]
,57E
1030 DATA 555DB6DD5555556DB6D5555555B6DB55 [2863]
,821
1040 DATA 5555576DB75555555B6DB00000018000 [1925]
,51D
1050 DATA 00036DB6AAAAAAADB6DAAAAAAAEDB6EA [2317]
,9EC
1060 DATA AAAAABB6DBAAAAAAAB6DB6AAAAAABB6D [2798]
,A82
1070 DATA B000001800000036DB75555555DB6DD [1862]
,547
1080 DATA 5555556DB6D5555555B6DB555555576D [2264]
,74A
1090 DATA B75555555B6DB000000180000036DB6 [1396]
,4D5
1100 DATA AAAAAAADB6DAAAAAAAEDB6EAAAAAABB6 [1213]
,B7B
1110 DATA DBAAAAAAAB6DB6AAAAAABB6DB0000001 [2516]
,87E
1120 DATA 800000036DB75555555DB6DD5555556D [1267]
,602
1130 DATA B6D5555555B6DB555555576DB7555555 [2088]
,794
1140 DATA 5B6DB0000001800000036DB6AAAAAAAD [2701]
,5CA
1150 DATA B6DAAAAAAAEDB6EAAAAAABB6DBAAAAAA [3490]
,BA9
1160 DATA AB6DB6AAAAAABB6DB00000180000003 [2030]
,628
1170 DATA 6DB75555555DB6DD5555556DB6D55555 [2011]
,7B4
1180 DATA 55B6DB555555576DB75555555B6DB000 [2150]
,6D7
1190 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1200 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1210 DATA AAAABB6DB000001800000036DB75555 [2229]
,57E
1220 DATA 555DB6DD5555556DB6D5555555B6DB55 [2863]
,821
1230 DATA 5555576DB75555555B6DB00000018000 [1925]
,51D
1240 DATA 00036DB6AAAAAAADB6DAAAAAAAEDB6EA [2317]
,9EC
1250 DATA AAAAABB6DBAAAAAAAB6DB6AAAAAABB6D [2798]
,A82
1260 DATA B000001800000036DB75555555DB6DD [1862]
,547
1270 DATA 5555556DB6D5555555B6DB555555576D [2264]
,74A
1280 DATA B75555555B6DB000000180000036DB6 [1396]
,4D5
1290 DATA AAAAAAADB6DAAAAAAAEDB6EAAAAAABB6 [1213]
,B7B
1300 DATA DBAAAAAAAB6DB6AAAAAABB6DB0000001 [2516]
,87E
1310 DATA 800000036DB75555555DB6DD5555556D [1267]
,602
1320 DATA B6D5555555B6DB555555576DB7555555 [2088]
,794
1330 DATA 5B6DB0000001800000036DB6AAAAAAAD [2701]
,5CA
1340 DATA B6DAAAAAAAEDB6EAAAAAABB6DBAAAAAA [3490]
,BA9
1350 DATA AB6DB6AAAAAABB6DB00000180000003 [2030]
,628
1360 DATA 6DB75555555DB6DD5555556DB6D55555 [2011]
,7B4
1370 DATA 55B6DB555555576DB75555555B6DB000 [2150]
,6D7
1380 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1390 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1400 DATA AAAABB6DB000001800000036DB75555 [2229]
,57E
1410 DATA 555DB6DD5555556DB6D5555555B6DB55 [2863]
,821
1420 DATA 5555576DB75555555B6DB00000018000 [1925]
,51D
```

Listing HANSEAT

```
1430 DATA 00036DB6AAAAAAADB6DAAAAAAAEDB6EA [2317]
,9EC
1440 DATA AAAAABB6DBAAAAAAAB6DB6AAAAAABB6D [2798]
,A82
1450 DATA B0000001800000036DB75555555DB6DD [1862]
,547
1460 DATA 5555556DB6D5555555B6DB555555576D [2264]
,74A
1470 DATA B75555555B6DB0000001800000036DB6 [1396]
,4D5
1480 DATA AAAAAAADB6DAAAAAAAEDB6EAAAAAABB6 [1213]
,B7B
1490 DATA DBAAAAAAAB6DB6AAAAAABB6DB0000001 [2516]
,87E
1500 DATA 800000036DB75555555DB6DD5555556D [1267]
,602
1510 DATA B6D5555555B6DB555555576DB7555555 [2088]
,794
1520 DATA 5B6DB0000001800000036DB6AAAAAAAD [2701]
,5CA
1530 DATA B6DAAAAAAAEDB6EAAAAAABB6DBAAAAAA [3490]
,BA9
1540 DATA AB6DB6AAAAAABB6DB000000180000003 [2030]
,628
1550 DATA 6DB75555555DB6DD5555556DB6D55555 [2011]
,7B4
1560 DATA 55B6DB555555576DB75555555B6DB000 [2150]
,6D7
1570 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1580 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1590 DATA AAAABB6DB0000001800000036DB75555 [2229]
,57E
1600 DATA 555DB6DD5555556DB6D5555555B6DB55 [2863]
,821
1610 DATA 5555576DB75555555B6DB00000018000 [1925]
,51D
1620 DATA 00036DB6AAAAAAADB6DAAAAAAAEDB6EA [2317]
,9EC
1630 DATA AAAAABB6DBAAAAAAAB6DB6AAAAAABB6D [2798]
,A82
1640 DATA B0000001800000036DB75555555DB6DD [1862]
,547
1650 DATA 5555556DB6D5555555B6DB555555576D [2264]
,74A
1660 DATA B75555555B6DB0000001800000036DB6 [1396]
,4D5
1670 DATA AAAAAAADB6DAAAAAAAEDB6EAAAAAABB6 [1213]
,B7B
1680 DATA DBAAAAAAAB6DB6AAAAAABB6DB0000001 [2516]
,87E
1690 DATA 800000036DB75555555DB6DD5555556D [1267]
,602
1700 DATA B6D5555555B6DB555555576DB7555555 [2088]
,794
1710 DATA 5B6DB0000001800000036DB6AAAAAAAD [2701]
,5CA
1720 DATA B6DAAAAAAAEDB6EAAAAAABB6DBAAAAAA [3490]
,BA9
1730 DATA AB6DB6AAAAAABB6DB000000180000003 [2030]
,628
1740 DATA 6DB75555555DB6DD5555556DB6D55555 [2011]
,7B4
1750 DATA 55B6DB555555576DB75555555B6DB000 [2150]
,6D7
1760 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1770 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1780 DATA AAAABB6DB0000001800000036DB75555 [2229]
,57E
1790 DATA 555DB6DD5555556DB6D5555555B6DB55 [2863]
,821
1800 DATA 5555576DB75555555B6DB00000018000 [1925]
,51D
1810 DATA 00036DB6AAAAAAADB6DAAAAAAAEDB6EA [2317]
,9EC
1820 DATA AAAAABB6DBAAAAAAAB6DB6AAAAAABB6D [2798]
,A82
1830 DATA B0000001800000036DB75555555DB6DD [1862]
,547
1840 DATA 5555556DB6D5555555B6DB555555576D [2264]
,74A
1850 DATA B75555555B6DB0000001800000036DB6 [1396]
,4D5
1860 DATA AAAAAAADB6DAAAAAAAEDB6EAAAAAABB6 [1213]
,B7B
```

Listing HANSEAT

```
1870 DATA DBAAAAAAAB6DB6AAAAAABB6DB0000001 [2516]
,87E
1880 DATA 800000036DB75555555DB6DD5555556D [1267]
,602
1890 DATA B6D5555555B6DB555555576DB7555555 [2088]
,794
1900 DATA 5B6DB0000001800000036DB6AAAAAAAD [2701]
,5CA
1910 DATA B6DAAAAAAAEDB6EAAAAAABB6DBAAAAAA [3490]
,BA9
1920 DATA AB6DB6AAAAAABB6DB000000180000003 [2030]
,628
1930 DATA 6DB75555555DB6DD5555556DB6D55555 [2011]
,7B4
1940 DATA 55B6DB555555576DB75555555B6DB000 [2150]
,6D7
1950 DATA 0001800000036DB6AAAAAAADB6DAAAAA [2993]
,736
1960 DATA AAEDB6EAAAAAABB6DBAAAAAAAB6DB6AA [2438]
,B3D
1970 DATA AAAABB6DB000000180000000FFFFFFFF [2902]
,7B8
1980 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
1990 DATA FFFFFFFFFFFFFFFFFFFFFC0000018000 [3035]
,B73
2000 DATA 00100000000000000000000000000000 [1343]
,010
2010 DATA 00000000000000000000000000000000 [2181]
,000
2020 DATA 02000001800000600000000000000000 [1890]
,0E3
2030 DATA 00000000000000000000000000000000 [2181]
,000
2040 DATA 00000000000001800001800000800000 [1744]
,182
2050 DATA 00000000000000000000000000000000 [2181]
,000
2060 DATA 00000000000000000000000000400001 [2184]
,041
2070 DATA 80000100000000000000000000000000 [946]
,081
2080 DATA 00000000000000000000000000000000 [2181]
,000
2090 DATA 00000200001800007FFFFFFFFFFFFFFF [2713]
,7A1
2100 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
2110 DATA FFFFFFFFFFFFFFFFFFF8000180001800 [2809]
,A88
2120 DATA 00000000000000000000000000000000 [2181]
,000
2130 DATA 00000000000000000000000000000006 [2286]
,006
2140 DATA 00018000200000000000000000000000 [2109]
,0A1
2150 DATA 00000000000000000000000000000000 [2181]
,000
2160 DATA 00000000000180018000C00000000000 [2061]
,1C2
2170 DATA 00000000000000000000000000000000 [2181]
,000
2180 DATA 000000000000000000000000C0018001 [2203]
,142
2190 DATA 00000000000000000000000000000000 [2181]
,000
2200 DATA 00000000000000000000000000000000 [2181]
,000
2210 DATA 00002001FFFFFFFFFFFFFFFFFFFFFFFF [2329]
,C15
2220 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
2230 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
2240 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
2250 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
2260 DATA 80000000000000000000000000000000 [1274]
,080
2270 DATA 00000000000000000000000000000000 [2181]
,000
2280 DATA 00000000000180000000000000000000 [1870]
,081
2290 DATA 00000000000000000000000000000000 [2181]
,000
2300 DATA C000000000000000000000180000000 0 [2220]
,141
2310 DATA 00000000000000000000000000000000 [2181]
,000
```

Listing HANSEAT

```
2320 DATA 000000000000C0000000000000000000 [1484]
,0C0
2330 DATA 00180000000000000000000000000000 [1584]
,081
2340 DATA 000000000000018000000000C0000000 [1081]
,141
2350 DATA 00000000000000018000000000000000 [1710]
,081
2360 DATA 0000000000000000000000000180003F [2131]
,0C0
2370 DATA FFFFFFFFFC0000000000000000018000 [2773]
,579
2380 DATA 00000000000000000000000000000000 [2181]
,000
2390 DATA 00000180000000012000000000000000 [1161]
,0A2
2400 DATA 00000001800000000000000000000000 [1774]
,081
2410 DATA 00000000000003FFFFFFFFFFFFFFFFFF [2230]
,936
2420 DATA FFF00000000000000018000000000000 [1780]
,270
2430 DATA 00000000000000000003000000000240 [1842]
,045
2440 DATA 01FFFFFFFFFFFF800000000000000001 [1980]
,67C
2450 DATA 80000000000000000000000000000003 [1296]
,083
2460 DATA 000FFFFFFFFFFDFFFFFFFFFFFF800000 [2918]
,B82
2470 DATA 00000000000180000000000000000000 [1870]
,081
2480 DATA 0000000000030001FFFFFFFFFDFFFFFF [2382]
,7FA
2490 DATA FFFFFFC0000000000000001800000000 [1699]
,43E
2500 DATA 000000000000000000003FFCFFFFFFFF [2639]
,5FA
2510 DATA FFFFFBFFFFFFFFFFFFFFFC0000000000 [1660]
,9B3
2520 DATA 00180000000000000000000000000000 [1584]
,081
2530 DATA 00048003FFFFFFFFFBFFFFFFFFFFFFE0 [2700]
,C58
2540 DATA 00000000000000018000000000000000 [1710]
,081
2550 DATA 000000000FFFFFFFFFFBFFFFFFFFFBFF [2044]
,AED
2560 DATA FFFFFFFFFE0000000000000000018000 [2802]
,65C
2570 DATA 000000000000000000000001FFFFFFF7 [1885]
,413
2580 DATA FFFFFFFFFBFFFFFFFFFFFFF000000000 [2317]
,BE1
2590 DATA 00000001800000000000000000000000 [1774]
,081
2600 DATA 001FFFFFFFEFFFFFFFFFF7FFFFFFFFFF [3163]
,DF9
2610 DATA FFF00000000000000018000000000000 [1780]
,270
2620 DATA 000000000000003FFFFFFDFFFFFFFFFF [2628]
,817
2630 DATA F7FFFFFFFFFFFF800000000000000001 [2796]
,7EA
2640 DATA 800000000000000000000000003FFFFF [1365]
,2BD
2650 DATA FFDFFFFFFFFFF7FFFFF24FFFFFF80000 [2984]
,D06
2660 DATA 00000000000180000000000000000000 [1870]
,081
2670 DATA 0000007FFFFFFFBFFFFFFFFFF7FFF002 [2590]
,B1F
2680 DATA 400FFFFC000000000000001800000000 [2350]
,2CB
2690 DATA 000000000000000000FFFFFFFFBFFFFF [2004]
,6B9
2700 DATA FA4FEFFC300240001FFC000000000000 [1809]
,4C1
2710 DATA 001800000000000000000000000000FF [1367]
,180
2720 DATA FFFFFF7FFFF86240E803F802400000 1E [1067]
,858
2730 DATA 00000000000000018000000000000000 [1710]
,081
2740 DATA 0000000001FFFFFF80FFF3FC1A400000 [1862]
,6C6
2750 DATA 00024000000000000000000000018000 [1879]
,0C3
```

Listing HANSEAT

```
2760 DATA 0000000000000000000001FFFE0481F8 [2524]
,37B
2770 DATA 0003FE40000000024000000000000000 [1854]
,183
2780 DATA 00000001800000000000000000000000 [1774]
,081
2790 DATA 03F00004800000024000000000024000 [2534]
,1FB
2800 DATA 00000000000000000001800000000000 [1894]
,081
2810 DATA 00000000000000000000480000000240 [2112]
,0C6
2820 DATA 00000002400000000000000000000001 [1359]
,043
2830 DATA 80000000000000000000000000000004 [1334]
,084
2840 DATA 80000000240000000240000000000000 [2064]
,104
2850 DATA 00000000000180000000000000000000 [1870]
,081
2860 DATA 0000000000480000000002 5FFFFFFFFF [2207]
,4E1
2870 DATA FFFFFFFC000000000000000180000000 [2173]
,53D
2880 DATA 00000000000000000000000480000000 [1984]
,084
2890 DATA 0243FFFFFFFFFFFFFFFE000000000000 [3283]
,83C
2900 DATA 00180000000000000000000000000000 [1584]
,081
2910 DATA 0004FFFFFFFFFBFFFFFFFFFFFFFFFFFF [2861]
,DF2
2920 DATA 00000000000000018000000000000000 [1710]
,081
2930 DATA 0000000000000004BFFFFFFFFFFBFFFF [2424]
,7B8
2940 DATA FFFFFFFFFFFF00000000000000018000 [2626]
,67B
2950 DATA 0000000000000000000000000004BFFF [1744]
,1C2
2960 DATA FFFFFFF7FFFFFFFFFFFFFFFFF0000000 [2176]
,BEC
2970 DATA 0000000180000000000000000000007F [1722]
,100
2980 DATA FFFFFFFFBFFFFFFFFF7FFFFFFFFFFFFF [3710]
,FA8
2990 DATA FFFF0000000000001800000000000000 [1083]
,27F
3000 DATA 0000000000FFFFFFFFFFF7FFFFFFFFF7 [2765]
,97D
3010 DATA FFFFFFFFFFFFFFF80000000000000001 [3047]
,879
3020 DATA 800000000000000000000000FFFFFFFF [1632]
,48B
3030 DATA 7FFFFFFFFFEFFFFFFFFFFFFFFFFF8000 [2670]
,DE2
3040 DATA 00000000000180000000000000000000 [1870]
,081
3050 DATA 001FFFFFFF7FFFFFFFFFFEFFFFFFFFFF [3399]
,D81
3060 DATA FFFFFFF8000000000000000180000000 [1523]
,4FD
3070 DATA 000000000000003FFFFFFFFEFFFFFC00 [1928]
,734
3080 DATA 3FEFFFFFFFFFFFFFFFFFC00000000000 [2806]
,9E6
3090 DATA 00180000000000000000000000 7FFFFF [1770]
,2FE
3100 DATA FFFEFFFE0000024FFFFE000240007FFF [1399]
,808
3110 DATA C0000000000000001800000000000000 [2141]
,141
3120 DATA 0000007FFFFC007EFF00000002400000 [1336]
,439
3130 DATA 00024000000000000000000000018000 [1879]
,0C3
3140 DATA 00000000000000000FF0000000048000 [1596]
,183
3150 DATA 00000240000000024000000000000000 [1632]
,084
3160 DATA 0000000180000000000000000C0000000 [2045]
,08D
3170 DATA 00000004800000000240000000024300 [1449]
,10B
3180 DATA 00000000000000000001800000000000 [1894]
,081
```

Listing HANSEAT

```
3190 DATA 000060000000000000004800000003FF [1919]
,18C
3200 DATA FFFFFFFFFFFFFFFFC000000000000001 [2102]
,8F5
3210 DATA 80000000000000003000000000000004 [1575]
,087
3220 DATA 80000000027FFFFFFFFFFFFFFFFFE000 [1942]
,9D9
3230 DATA 00000000000180000000000000000180 [2142]
,102
3240 DATA 0000000001FFFFFFFFFFFF7FFFFFFFFF [2000]
,A76
3250 DATA FFFFFFFFE00000000000000180000000 [2311]
,55D
3260 DATA 0000000000C01FFFFFFFFDFFFFFFFFFF [1808]
,9D4
3270 DATA FF7FFFFFFFFFFFFF3FFF000000000000 [1773]
,A5A
3280 DATA 000180000000000000000046007FFFFFF [2521]
,3E9
3290 DATA FDFFFFFFFFFFFF7FFFFFFFFFFFFFADFF [1829]
,F1C
3300 DATA F8000000000000018000000000000000 [2062]
,179
3310 DATA 0E3007FFFFFFFBFFFFFFFFFFFEFFFFFF [2138]
,D33
3320 DATA FFFFFFFF96FFFC000000000000018000 [2112]
,70E
3330 DATA 0000000000000F180FFFFFFFFBFFFFFF [2633]
,72B
3340 DATA E001FEFFFFFFF002407F5B7FFE000000 [1234]
,865
3350 DATA 000000018000000000000000001F8C0FFF [2019]
,23A
3360 DATA FFFE03FF800000006FF80000000024000 [1291]
,546
3370 DATA ED8FFF00000000000001800000000000 [1433]
,2FC
3380 DATA 00003FE61FF800000004800000000240 [1985]
,302
3390 DATA 000000024000EDE0000000000000000001 [1783]
,210
3400 DATA 80000000000000007FF18000000000004 [1786]
,274
3410 DATA 800000000240000000024001EDF00000 [1379]
,2E2
3420 DATA 00000000000180000000000000007FF8 [2053]
,1F8
3430 DATA C0000000000480000003FFFFFFFFFFF [2883]
,67E
3440 DATA FFFDEEFDFF000000000000018000000 [1776]
,567
3450 DATA 00000000FFFC600000001FFFFFFFFFFF [1733]
,775
3460 DATA FBFFFFFFFFFFFFFBEEFEFF0000000000 [1399]
,ADB
3470 DATA 00018000000000000001FFFE30000000 [1771]
,2AF
3480 DATA 03FFFFFFFFFFFBFFFFFFFFFFFFFFBEF7F [2815]
,E5C
3490 DATA 7F800000000000018000000000000003 [2107]
,183
3500 DATA FFFF7FFFFFFFFBFFFFFFFFFFFBFFFFFF [2020]
,F68
3510 DATA FFFFFFF7DF7FDF800000000000018000 [2198]
,732
3520 DATA 00000000003FFFF7FFFFFFFFFBFFFFFF [2466]
,975
3530 DATA FFFFF7FFFFFFFFFFFFF7DFBFEFC00000 [2440]
,D33
3540 DATA 00000001800000000000000007FFFEFFFF [2050]
,483
3550 DATA FFFFF7FFFFFFFFFFF7FFFFFFFFFFFFEF [2832]
,FD0
3560 DATA DFBFF7C00000000000001800000000000 [2357]
,3D6
3570 DATA 000FFFFEFFFFFFFF7FFFFFFFFFFFF7FF [1532]
,DF0
3580 DATA FFFFFFFFFFEFDFBFFDE000000000001 [2936]
,966
3590 DATA 80000000000001FFFFDFFFFFFFFEFFF [1494]
,885
3600 DATA FFFFFFFFF7FFFFFFFFFFFFDFDFDFFEE0 [1937]
,F68
3610 DATA 00000000000180000000000001FFFFD [2238]
,29C
```



```
3620 DATA FFFFFFFFEFFFFFFFFFFFF7FFFFFFFFFF [2367]
,FD8
3630 DATA FFDFDFDFFF70000000000000180000000 [2122]
,58C
3640 DATA 0000003FFFFBFFFFFFFFDFFFFFFFFF80 [2098]
,B90
3650 DATA 0FFFFFFFFFFFFFBFBFEFFFD000000000 [3526]
,A45
3660 DATA 00018000000000000007FFFFBFFFF8000 [1547]
,578
3670 DATA 1FFFFFFC00001FFFFFFFFFFFFFBFBFE3 [1810]
,B93
3680 DATA FFE800000000000018000000000000FF [2350]
,367
3690 DATA FFF8018000003FFFE00000001FFFFFFF [2849]
,7B2
3700 DATA F801207F8FFC3FF000000000000018000 [1905]
,4D3
3710 DATA 000000000FFFFF7F18000007F048000 [1779]
,569
3720 DATA 00003FFFFE000001201FF0FFC3FC0000 [2006]
,62A
3730 DATA 0000000180000000000001FFFFEFF180 [1593]
,4E0
3740 DATA 0000800480000007FC000000000012001 [1537]
,265
3750 DATA FF0FFC32000000000000180000000000 [1694]
,2BD
3760 DATA 001FFFEFF18000008004800000004240 [1411]
,504
3770 DATA 000000012000 3FF0FFCFC0000000001 [3147]
,3DF
3780 DATA 8000000000000003FDFF18000010004 [1145]
,314
3790 DATA 80000008240000000001200007FF0F46 [1202]
,2BE
3800 DATA 0000000000018000000000000007C00BF [2088]
,1BC
3810 DATA F1800001000480000000082400000001 [1863]
,2B9
3820 DATA 2000007FF3F800000000000180000000 [2272]
,30B
3830 DATA 00000043FF41F1800002000480000000 [1343]
,37A
3840 DATA 82400000000012000FFCFFFC000000000 [1532]
,470
3850 DATA 00018000000000000007C013FFD800004 [1613]
,2BE
3860 DATA 000480000001024000001FFFFFFF003F [1966]
,422
3870 DATA 03000000000000018000000000000023 [1779]
,0A7
3880 DATA FFC003FFFE040004800000013FFFFFFF [2073]
,784
3890 DATA E0000000FFC0040000000000000018000 [1842]
,324
3900 DATA 000000000010003FFC0001FFFFFC8000 [2150]
,4C6
3910 DATA 7FFFC00000001FFFFFFF000008000000 [2928]
,562
3920 DATA 00000001800000000000000086000 03FF [2556]
,1EB
3930 DATA FE000007FFFF80003FFFFFFFE0000000 [2280]
,79F
3940 DATA 60803000000000000001800000000000 [1645]
,191
3950 DATA 0008060600001FFFFF800007FFFC000 [1917]
,5FD
3960 DATA 00000606060088840000000000000001 [2336]
,1F1
3970 DATA 80000000000000400000060600000007 [1664]
,14B
3980 DATA FFFF8000006060606000000004908000 [1400]
,512
3990 DATA 00000000000180000000000000020000 [2078]
,083
4000 DATA 00000606060000000606000000000000 [2607]
,1E0
4010 DATA 00003610000000000000000180000000 [1948]
,0E5
4020 DATA 00000001000000000000000060606000 [1669]
,121
4030 DATA 00000000000000000020000000000000 [1826]
,002
4040 DATA 00018000000000000000800000000000 [1654]
,101
```

```
4050 DATA 00000000000000000000000000000004 [2003]
,004
4060 DATA 000000000000000180000FFC00000000 [1978]
,18C
4070 DATA 9FF00000000000000000000000000000 [1769]
,18F
4080 DATA 00003FFE00080000000003FE0000181FF [1878]
,3E5
4090 DATA F557F800001FF55FF000000000000FFF [2262]
,5B5
4100 DATA FC0000000003FFFD555FF900000000FF [1902]
,5F2
4110 DATA D55FE001FEAAAAAAAFE01FEAAAAAAFF0 [820]
,A9C
4120 DATA 000FFFFFFAAAABFFE0003FFFEAAAAAAA [2050]
,A61
4130 DATA AAFFE003FFAAAAAABFE1D5555555555F [2105]
,9B1
4140 DATA F5555555555FFFF55555555555555FFF [1931]
,7F8
4150 DATA D555555555555555FFD555555555555F [2231]
,68C
4160 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
4170 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
4180 DATA AAAAAAAAAABD55555555555555555555 [2853]
,7CF
4190 DATA 55555555555555555555555555555555 [2522]
,550
4200 DATA 5555555555555555555555555AAAAAAA [2130]
,6A4
4210 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
4220 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
4230 DATA AAABD555555555555555555555555555 [2408]
,67B
4240 DATA 55555555555555555555555555555555 [2522]
,550
4250 DATA 5555555555555555AAAAAAAAAAAAAAAA [2266]
,7F8
4260 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
4270 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAABFFFF [2119]
,B4B
4280 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
4290 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
4300 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
4310 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
4320 DATA FFFFFFFFFFFFFFFFFFFF800000000000 [2485]
,A76
4330 DATA 00000000000000000000000000000000 [2181]
,000
4340 DATA 00000000000000000000000000000001 [2168]
,001
4350 DATA 80000000000000000000000000000000 [1274]
,080
4360 DATA 00000000000000000000000000000000 [2181]
,000
4370 DATA 00000000000180000000000000000000 [1870]
,081
4380 DATA 00000000000000000000000000000000 [2181]
,000
4390 DATA 00000000000000000000000180000000 [1862]
,081
4400 DATA 00000000000000000000000000000000 [2181]
,000
4410 DATA 00000000000000000000000000000000 [2181]
,000
4420 DATA 00018000000000000000000000000000 [1584]
,081
4430 DATA 00000000000000000000000000000000 [2181]
,000
4440 DATA 00000000000000018000000000000000 [1710]
,081
4450 DATA 00000000080000000000000000000000 [1702]
,008
4460 DATA 00000000000000000000000000018000 [2115]
,081
```

Listing HANSEAT

```
4470 DATA 00000000000000000001400000000000 [1969]
,014
4480 DATA 00000020204000000000000000000000 [1479]
,080
4490 DATA 00000001800000000000000000000000 [1774]
,081
4500 DATA 2A0000000001000003811D7C00000000 [1958]
,148
4510 DATA 00000000000000000001800000000000 [1894]
,081
4520 DATA 00000000000055000004009134003FB3 [1948]
,210
4530 DATA 39DC4810000000000000000000000001 [2232]
,16E
4540 DATA 800000000000000000000000AB000024 [1642]
,14F
4550 DATA 048FFB03FFFB7BEE5800000000000000 [1873]
,54C
4560 DATA 00000000000180000000000000000000 [1870]
,081
4570 DATA 0000D580005944EFBB93F3FBDFFFE4C0 [2819]
,99F
4580 DATA 00000000000000000000000180000000 [1862]
,081
4590 DATA 0000000000000001AAC00170A76C9E81 [1902]
,40E
4600 DATA FFA37FFFEEA200000000000000000000 [1445]
,4B0
4610 DATA 00018000000000000000000000035560 [1475]
,139
4620 DATA 0000026D681055FFEFFFA5C303800000 [1895]
,614
4630 DATA 00000000000000018000000000000000 [1710]
,081
4640 DATA 00000006AAB00000009F3C48D7FFFFFF [2004]
,657
4650 DATA 4BAE1AD0240022200000000000018000 [2307]
,2CA
4660 DATA 00000000000000000000D5580000000B [1394]
,0C5
4670 DATA 2A003B633EED1055C1116FC540117028 [2549]
,547
4680 DATA 8008000180000000000000000000001A [1788]
,123
4690 DATA AAAC000000101020010010E088908843 [2148]
,46A
4700 DATA 7F8D7999882002000001800000000000 [2043]
,349
4710 DATA 00000000035555400000000000000000 [2094]
,0DE
4720 DATA 000008880C819C12004B0CA081000001 [2136]
,344
4730 DATA 80000000000000000000006AAAAA0000 [2149]
,23E
4740 DATA 00000000000000000080900070B00000 [2458]
,230
4750 DATA 00000000000180000000000000000000 [1870]
,081
4760 DATA 00D55550000000000000000000000000 [1412]
,17F
4770 DATA 00000000000000000000000180000000 [1862]
,081
4780 DATA 0000000000001AAAAA80000000000000 [1788]
,27F
4790 DATA 00000000000000000000000000000000 [2181]
,000
4800 DATA 00018000000000000000000001555555 [1619]
,181
4810 DATA 40000000000000000000000000000000 [1446]
,040
4820 DATA 00000000000000018000000000000000 [1710]
,081
4830 DATA 000002AAAAAA00000000000000000000 [2072]
,2A0
4840 DATA 00000000000000000000000000018000 [2115]
,081
4850 DATA 00001C00000000005555555500000000 [1274]
,170
4860 DATA 00180000000000000000000000000000 [2079]
,018
4870 DATA 00000001800000006B800000000000AAA [2358]
,220
4880 DATA AAAAA8000000007E0000000000000000 [1621]
,27A
```

Listing HANSEAT

```
4890 DATA 0000000000000000000180000001D560 [1905]
,1B7
4900 DATA 00000001555555558000000001998000 [2146]
,286
4910 DATA 00000000000000000000000000000001 [2168]
,001
4920 DATA 80000006AABC000000002AAAAAAAC00 [2174]
,4C0
4930 DATA 00000618600000000000000000000000 [2054]
,07E
4940 DATA 000000000018000003D555780000000 [2086]
,1EA
4950 DATA 55555555560000001818180000000000 [2187]
,1F2
4960 DATA 000000000000000000000001800000EA [1986]
,16B
4970 DATA AAAAE0000000AAAAAAAAB0000006018 [2132]
,5FF
4980 DATA 07FFFFFFFFFF00000000000000000000 [1782]
,502
4990 DATA 0001800003555555C000001555555555 [2356]
,334
5000 DATA 5580000180180 1D555555555580000000 [2143]
,418
5010 DATA 000000000000000180000FFFFFFFFF00 [2543]
,48C
5020 DATA 0002AAAAAAAAAAC00007FFFFFFEAAAAA [1105]
,956
5030 DATA AAAA8000000000000000000000018000 [2663]
,255
5040 DATA 30000000003000003555555555600019 [2504]
,258
5050 DATA 007E009D5555555540000000000000000 [1827]
,2AF
5060 DATA 000000018000C000000000C00006AAAA [1893]
,35B
5070 DATA AAAAAAA0006101998086AAAAAAAAA000 [2336]
,7E7
5080 DATA 0000000000000000000018003003FFFFE [2054]
,2C0
5090 DATA 0030000D555555555555001810618608 1 [2416]
,3B7
5100 DATA D555555563FFFF80003FFFF80003FFFF [2129]
,8EC
5110 DATA 800C00222222000C001AAAAAAAAAAAA8 [2751]
,512
5120 DATA 060118181880 6AAAAAAAB3BBBB80003B [2074]
,61B
5130 DATA BBB80003BBBB8010002222222 0003003F [1419]
,424
5140 DATA FFFFFFFFFFFC180160180680 1D555555 [2880]
,82A
5150 DATA 5AEEEE80002EEEE80002EEEF80600022 [2585]
,79B
5160 DATA 22220000802000000000000860018018 [1917]
,1E5
5170 DATA 018006AAAAAAAF77778000377778 0003 [2033]
,5CB
5180 DATA 777781800022222200006027FE01FF80 [1354]
,55A
5190 DATA FFC9FFFFFFFFFFFFFFD555555 7DDDD80 [1624]
,CD1
5200 DATA 003DDDD80003DDDD8600002222220000 [1656]
,49B
5210 DATA 1824620118808C47FFFFFFFFFFFFFFFF [2237]
,A02
5220 DATA FFFFFF77778000377778000377779800 [1782]
,71A
5230 DATA 003FFFFE0000062462011880 8C4C0000 [2102]
,439
5240 DATA 0000004000000000007DDDD80003DDDD8 [2770]
,473
5250 DATA 0003DDDDE00000000000000001A7FE01 [3205]
,444
5260 DATA FF80FFCCFFFE00FFFE4FFFE00FFFE777 [2140]
,BDE
5270 DATA 77800037777800037777FFFFFFFFFFFF [2870]
,908
5280 DATA FFFFFFE4620118808C4C810200810248 [1496]
,702
5290 DATA 1020081027DDDD80003DDDD80003DDDD [2147]
,658
5300 DATA 800006000030000006462011880 8C4C [1424]
,31A
5310 DATA 810200810248102008102777 77FFFFF7 [1564]
,5A0
```

Listing HANSEAT

```
5320 DATA 777FFFFF777783FFFF9000047FFFF064 [2279]
,9C9
5330 DATA 620118808C4C810200810248102008 10 [2229]
,369
5340 DATA 27DDDDDDDDDDDDDDDDDDDDDD8201008C [3808]
,AB5
5350 DATA 001840101067FE01FF80FFCCFFFE00FF [1542]
,824
5360 DATA FE4FFFE00FFFE6EEEEEEEEEEEEEEEEE [2601]
,D7E
5370 DATA EEEF820101FFFFFFC010106606018180 [1976]
,7AC
5380 DATA C0CC8102008102481020081027BBBBBB [1381]
,57A
5390 DATA BBBBBBBBBBBBBBB8201008180C04010 [1660]
,86C
5400 DATA 10619800660033 0C8102008102481020 [1371]
,32C
5410 DATA 0810277777777 7FFFF77777777783FF [3047]
,7EE
5420 DATA FF8041007FFFF060600018000C0C8102 [1518]
,5A1
5430 DATA 0081024810200810 27DDDDDDFE0007F [2445]
,60F
5440 DATA DDDDDDD820100803600401010619800 [2618]
,606
5450 DATA 6600330CFFFE00FFFE4FFFE00FFFE7BB [1480]
,97D
5460 DATA BBBBE0000007BBBBBBB820100800800 [2261]
,60D
5470 DATA 40101066060181 80C0CCC00600C0064C [2306]
,532
5480 DATA 00600C0066EEEEEF80000001EEEEEEF [1795]
,706
5490 DATA 820100803600401 0107FFFFFFFFFFFC [1601]
,80F
5500 DATA 301800301843018003018 7BBBBB80000 [1615]
,40D
5510 DATA 000001BBBBBB820100FFFFFFC0101060 [1905]
,6F2
5520 DATA 0000000000C0C60000C6040C60000C6 [2273]
,2B0
5530 DATA 06EEEFE000000000007EEEEFFFFFFFFF [3020]
,91A
5540 DATA FFFFFFFFFFE7FE01FF80FFCC03800003 [2891]
,AB1
5550 DATA 80403800003807777780000000000 01F [1827]
,2C4
5560 DATA 77778303C0800000607810646201188 0 [1366]
,4FB
5570 DATA 8C4C06C00006C0406C00006C07DDDE00 [2324]
,53E
5580 DATA 0000000000007DDDD80CC33000000 1986 [2343]
,3DF
5590 DATA 6064620118808C4C183000183041830 0 [1825]
,3EB
5600 DATA 0183077776000000000000077777 8030 [2312]
,31D
5610 DATA 0C0000000601806 7FE01FF80FFCC600C [2070]
,5AF
5620 DATA 00600C4600C00600C7DDDE0000000000 [1008]
,3FA
5630 DATA 0007DDDD80CC330000000198660646201 [2095]
,506
5640 DATA 18808C4FFFFFFFFFFFFFFFFFFFFFFF77 [2724]
,CDF
5650 DATA 7600000000000007777 78303C0800000 [2478]
,331
5660 DATA 6078106462011880 8C4C8A288000004 0 [2189]
,491
5670 DATA 0000000007DDDE00000000000000 7DDDD [2532]
,383
5680 DATA FFFFFFFFFFFFFFFFFFFE4620118808C4C [3369]
,BAE
5690 DATA 514500FFFE4FFFE00FFFE77776000000 [2601]
,7A3
5700 DATA 0000007777780000060000300000067 [2097]
,23F
5710 DATA FE01FF80FFCCFFFF808102481020081 0 [1416]
,7DA
5720 DATA 27DDDE0000000000007DDDD83FFFF90 [2369]
,6B4
5730 DATA 00047FFFF066060181 80C0CC51450081 [1823]
,683
5740 DATA 02481020081026EEEE000000000000006 [2524]
,29A
```

Listing HANSEAT

```
5750 DATA EEEF8201008C0018401010619800660 0 [1834]
```

```
5750 DATA EEEF8201008C00184010106198006600 [1834]
,4C3
5760 DATA 330C8A288081024810200810 27BBBE00 [1938]
,424
5770 DATA 000000000007BBBB820101FFFFFFC010 [2159]
,5CE
5780 DATA 1060600018000C0C014000FFFE4FFFE0 [3334]
,56C
5790 DATA 0FFFE777760000000000000777778201 [1591]
,45A
5800 DATA 008180C04010106198006600330C0550 [1251]
,414
5810 DATA 0081024810200810 27DDDE0000000000 [1651]
,2F5
5820 DATA 0007DDDD83FFFF8041007FFFF0660601 [2164]
,7DE
5830 DATA 8180C0CC15540081024810200810 27BB [2314]
,4EB
5840 DATA BE00000000000007BBBB820100803600 [2086]
,374
5850 DATA 4010107FFFFFFFFFFFFC155400810248 [2806]
,80A
5860 DATA 1020081026EEEE00000000000006EEEF [2507]
,42D
5870 DATA 8201008008004010106000000000000C [2319]
,1D7
5880 DATA 555500FFFE4FFFE00FFFE7BBBE000000 [2974]
,843
5890 DATA 00000007BBBB82010080360040101060 [2586]
,376
5900 DATA F083FFC1FFCC555500C0064C00600C00 [2285]
,726
5910 DATA 66EEEE00000000000006EEEF820100FF [1680]
,5A7
5920 DATA FFFFC0101063FC4518A28C4C55550030 [1931]
,6EE
5930 DATA 18430180030187777600000000000007 [2161]
,25B
5940 DATA 7777FFFFFFFFFFFFFFFFFFE7FE291894 [3117]
,C9F
5950 DATA 8C4C5555000C6040C60000C607DDDE00 [2931]
,57C
5960 DATA 000000000007DDDD8303C08000006078 [2101]
,45F
5970 DATA 1067FE11FF88FFCC7FFF000380403800 [2385]
,751
5980 DATA 003807777600000000000007777780CC [1635]
,36D
5990 DATA 3300000019866067FE2918948C4C7FFF [1991]
,5C2
6000 DATA 0006C0406C00006C07DDDE0000000000 [2267]
,3A0
6010 DATA 0007DDDD80300C000000060180647E45 [2259]
,42B
6020 DATA 18A28C4C555500183041830001830777 [1581]
,44A
6030 DATA 7600000000000007777780CC33000000 [2063]
,2EA
6040 DATA 19866067FE83FFC1FFCC555500600C46 [1854]
,7CE
6050 DATA 00C00600C7DDDE00000000000007DDDD [1234]
,509
6060 DATA 8303C080000060781067FE000000000C [1317]
,41F
6070 DATA 55550000004000000000077776000000 [2024]
,1DE
6080 DATA 000000077777FFFFFFFFFFFFFFFFFFE7 [1775]
,AD3
6090 DATA FE000000000C55550000004000000000 [1368]
,1F4
6100 DATA 07DDDE00000000000007DDDDFFFFFFFF [1384]
,77F
6110 DATA FFFFFFFFFFFFFFFFFFFFFFFFD555FFFF [2978]
,F1C
6120 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
6130 DATA FFFFAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2427]
,B4A
6140 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6150 DATA AAAAAAAAAAAAAAABD555555555555555 [2751]
,879
6160 DATA 55555555555555555555555555555555 [2522]
,550
6170 DATA 5555555555555555555555555555AAAA [1869]
,5FA
6180 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6190 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
```

Listing HANSEAT

```
6200 DATA AAAAAAABD55555555555555555555555 [880]
,725
6210 DATA 55555555555555555555555555555555 [2522]
,550
6220 DATA 55555555555555555555AAAAAAAAAAAA [3082]
,74E
6230 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6240 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAB [2224]
,AA1
6250 DATA D5555555555555555555555555555555 [2329]
,5D0
6260 DATA 55555555555555555555555555555555 [2522]
,550
6270 DATA 555555555555AAAAAAAAAAAAAAAAAAAA [3094]
,8A2
6280 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6290 DATA AAAAAAAAAAAAAAAAAAAAAAABD5555555 [1864]
,9CD
6300 DATA 55555555555555555555555555555555 [2522]
,550
6310 DATA 55555555555555555555555555555555 [2522]
,550
6320 DATA 5555AAAAAAAAAAAAAAAAAAAAAAAAAAAA [2314]
,9F6
6330 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6340 DATA AAAAAAAAAAAAAAABFFFFFFFFFFFFFFFF [2851]
,D49
6350 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
6360 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
6370 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
6380 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
6390 DATA FFFFFFFFAAAAAAAAAAAAAAAAAAAAAB00 [3323]
,B4B
6400 DATA 0000000007AAAC0000000000000006AA [1395]
,20D
6410 DATA AAAAAAB80001FAAAAAABD55555555555 [1373]
,82E
6420 DATA 555555555580000000003D5556000000 [2408]
,311
6430 DATA 000000000D5555555556003F55555555 [2191]
,34A
6440 DATA AAAAAAAAAAAAAAAAAAAAAA8000000001 [2569]
,7CF
6450 DATA EAAAAB000000000000000EAAAAAAAAAB [1457]
,5A0
6460 DATA 87EAAAAAAAABD5555555555555555555 [1710]
,7EC
6470 DATA 55700000001F555555C0000000000000 [1138]
,2A3
6480 DATA 01D5555555557D5555555555AAAAAAAA [2054]
,6F8
6490 DATA AAAAAAAAAAAAAAAFC00000EAAAAAAABE [3072]
,9BB
6500 DATA 000000000000006AAAAAAAAAAAAAAAAA [859]
,5BA
6510 DATA AAABD555555555555555555555557E00 [2375]
,64F
6520 DATA 075555555556000000000000000D55555 [1532]
,330
6530 DATA 5555555555555555AAAAAAAAAAAAAAAA [2266]
,7F8
6540 DATA AAAAAAAAABF87AAAAAAAAAAE00000000 [3563]
,81B
6550 DATA 00000FAAAAAAAAAAAAAAAAABFFFFD555 [1677]
,932
6560 DATA 555555555555555555555557D5555555 [2329]
,5D2
6570 DATA 5557000000000001F555555555555555 [2572]
,3F5
6580 DATA 557E0001AAAAAAAAAAAAAAAAAAAAAAAA [2108]
,8CC
6590 DATA AAAAAAAAAAAAAAAB000000000002AAAA [3463]
,6A7
6600 DATA AAAAAAAAAAAABF800001D55555555555 [2350]
,7BA
6610 DATA 55555555555555555555555555550000 [2555]
,4A6
6620 DATA 000000055555555555555555E0000001 [1797]
,38E
6630 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA [2236]
,AA0
6640 DATA AAAAAAAA80000000000AAAAAAAAAAAAA [2839]
,72E
```

Listing HANSEAT

```
6650 DATA AAAB00000001D5555555555555555555 [1449]
,528
6660 DATA 555555555555F5555554000000000015 [2089]
,3B1
6670 DATA 57D5555555555555500000001AAAAAAA [2007]
,5D3
6680 DATA AAAAAAAAAAAAAAAAAAAAAAAAA2AAAAAA [1785]
,A98
6690 DATA B0000000002AA86AAAAAAAAAAAAA8000 [2284]
,668
6700 DATA 0001D555555555555555555555555555 [2749]
,527
6710 DATA 5555435555555C000000000355855555 [2230]
,3D4
6720 DATA 555555557F000001AAAAAAAAAAAAAAAA [2486]
,724
6730 DATA AAAAAAAAAAAAAAAA82AAAAAAAF000000 [3234]
,87F
6740 DATA 006EA86AAAAAAAAAAAAAAA800001D555 [2201]
,7D1
6750 DATA 5555555555555555555555557FFF0355 [2051]
,5D2
6760 DATA 555555C0000000DD57D5555555555555 [2057]
,5C6
6770 DATA 55800001AAAAAAAAAAAAAAAAAAAAAAAA [1359]
,8CE
6780 DATA AAAAC00006AAAAAAAAB0000000AEAAAA [3251]
,774
6790 DATA AAAAAAAAAAAAAA800001D55555555555 [2068]
,7A5
6800 DATA 55555555555555555FFFFD555555555C [2741]
,6B3
6810 DATA 000000DD5555555555555D5555800001 [1309]
,463
6820 DATA AAAAAAAAAAAAAAAAAAAAAAAAAAAAF013 [2446]
,A4F
6830 DATA AAAAAAAAAAAC000001BAAAAAAAAAAAAA [2792]
,8B5
6840 DATA B3EAAA800001D5555555555555555555 [1851]
,69A
6850 DATA 5555555F09E0D5555555555540000015D [2241]
,522
6860 DATA 555555555555401FD5800001AAAAAAAA [2827]
,65B
6870 DATA AAAAAAAAAAAAAAAAABE006203AAAAAAA [2103]
,939
6880 DATA AAAC000002BAAAAAAAAAAAAA80007E80 [2502]
,78C
6890 DATA 0001D555555555555555555555557C00 [2366]
,4F9
6900 DATA 00000F555555555540000037555555555 [2086]
,383
6910 DATA 55550000003800001AAAAAAAAAAAAAAA [2602]
,67E
6920 DATA AAAAAAABC000000003AAAAAAAAAA0000 [2476]
,6BE
6930 DATA FEAAAAAAAAAAAAAA000000000001D555 [1919]
,6CF
6940 DATA 555555555555555555558000000000FD [2108]
,4CF
6950 DATA 55555555000155555555555555540000 [2347]
,3FC
6960 DATA 0000001AAAAAAAAAAAAAAAAAAAAAAAAA [2527]
,7F9
6970 DATA C0000000006AAAAAAAA8001AAAAAAAA [1622]
,697
6980 DATA AAAAAAA80000000000001D5555555555 [2164]
,525
6990 DATA 555555555D78000000003CD55555555 [2115]
,524
7000 DATA 7FFF55555555555555550000000000001 [3051]
,422
7010 DATA AAAAAAAAAAAAAAAAAAAAABAA00000000 [2536]
,7F9
7020 DATA 02BAAAAAAFAAAAAABEAAAAAAAAAAAAA0 [1957]
,A17
7030 DATA 00000000001D55555555555555555555 [1871]
,3D3
7040 DATA 5554000000000355557FF1D555555F55 [2274]
,4F9
7050 DATA 55555555555600000000000001AAAAAAA [3515]
,4B2
7060 DATA AAAAAAAAAAAAAAAB000000003EEAABC0 [2685]
,7E4
7070 DATA 02AAAAAAAAAAAAAAAAAAAAB000000000 [2449]
,756
7080 DATA 0001D55555555555555555555550000 [2716]
,47D
7090 DATA 00005555560005555555555555555555 [1333]
,402
```

Listing HANSEAT

```
7100 DATA 55500000000000001AAAAAAAAAAAAAAA [1260]
,5F6
7110 DATA AAAAABAB000000006AAAAA0006AAAAAA [1909]
,66C
7120 DATA AAAAAAAAAAAAAAB0000000000001D555 [2957]
,681
7130 DATA 55555555555555557D40000003D555 [2401]
,500
7140 DATA 55000D5555555555555555555580000 [2106]
,40C
7150 DATA 0000001AAAAAAAAAAAAAAAAAAAAAAAAE [3040]
,7FD
7160 DATA 0000006AAAAAB007AAAAAAAAAAAAAAA [2841]
,725
7170 DATA AAAAAAA80000000000001D5555555555 [2164]
,525
7180 DATA 55555555555600000005555557F5FFD555 [2408]
,5CE
7190 DATA 5555555555555555555540000000000001 [1782]
,352
7200 DATA AAAAAAAAAAAAAAAAAAAAAAAE0000001E [2000]
,81A
7210 DATA AAA81AAAAAAAAAAAAAAAAAAAAAAABFFC [2075]
,A75
7220 DATA 00000000001D55555555555555555555 [1871]
,3D3
7230 DATA 5750000000155550D575555555555555 [1877]
,3C3
7240 DATA 55555555400000000000000001AAAAAAAA [2141]
,43D
7250 DATA AAAAAAAAAAAAABB00000000EAAA81AAC [2714]
,77D
7260 DATA FAAAAAAAAAAAAAAAAAAAC00000000000 [1478]
,7B4
7270 DATA 0001D55555555555555555555555600000 [2595]
,488
7280 DATA 000D5557F558075555555555555555555 [2792]
,50A
7290 DATA 6000000000000001AAAAAAAAAAAAAAAA [1804]
,5B1
7300 DATA AAAAABA00000000EAAAAAABFFFAAAAAA [1285]
,867
7310 DATA AAAAAAABEAAAA0000000000000001D555 [2353]
,608
7320 DATA 55555555555555555760000000015555 [2491]
,40A
7330 DATA 55555555555555555555E1F55C0000000 [2572]
,48F
7340 DATA 0000001AAAAAAAAAAAAAAAAAAAAAABA0 [2626]
,7F0
7350 DATA 0000002AAAAAAAAAAAAAAAAAAAAAAAE0 [2228]
,830
7360 DATA 01FB0000000000000001D55555555555 [1099]
,37B
7370 DATA 55555555554000000001555555555555 [2342]
,3E8
7380 DATA 5555555557000000E000000000000001 [2335]
,1BA
7390 DATA AAAAAAAAAAAAAAAAAAAAAEC000000000 [2459]
,812
7400 DATA EAAAAAAAAAEAAAAAAAAAF80000000000 [1783]
,81C
7410 DATA 00000000001D55555555555555555555 [1871]
,3D3
7420 DATA 5D40000000003FF5555555D555555557 [2595]
,4FB
7430 DATA 800000000000000000000001AAAAAAAA [1172]
,329
7440 DATA AAAAAAAAAAAAAA800000000001AAAAA [2005]
,694
7450 DATA BEBAAAAAAABC00000000000000000000 [1952]
,432
7460 DATA 0001D5555555555555555555575800000 [2736]
,4C8
7470 DATA 00000055555E1F5555555E000000000 [2274]
,464
7480 DATA 0000000000000001AAAAAAAAAAAAAAAA [1195]
,551
7490 DATA AAAABB0000000000006AABE001FAAAA [1578]
,4F0
7500 DATA AE0000000000000000000000001D555 [1270]
,1D9
7510 DATA 5555555555555555C000000000000003 [1621]
,307
7520 DATA D5E00000FD557000000000000000000 [1813]
,377
7530 DATA 0000001AAAAAAAAAAAAAAAAAAAAB000 [2758]
,755
7540 DATA 000000000007E0000007AF80000000 [2220]
,1B4
```

Listing HANSEAT

```
7550 DATA 00000000000000000001D55555555555 [2325]
,27F
7560 DATA 55555555600000000000000020000000 [2599]
,1D4
7570 DATA 00780000000000000000000000000001 [1924]
,079
7580 DATA AAAAAAAAAAAAAAAAAAAF800000000000 [2553]
,729
7590 DATA 00000000000000000000000000000000 [2181]
,000
7600 DATA 0000000001D5555555555555D5D57F8 [2012]
,488
7610 DATA 00000000000000000000000000000000 [2181]
,000
7620 DATA 0000000000000000000000001AAAAAAA [2738]
,2A9
7630 DATA AAAAEAAAB80000000000000000000000 [1565]
,3A0
7640 DATA 00000000000000000000000000000000 [2181]
,000
7650 DATA 0001D55F7FDF5DD557FD400000000000 [2841]
,559
7660 DATA 00000000000000000000000000000000 [2181]
,000
7670 DATA 0000000000000001AABAAAAAAAAFF806 [2226]
,510
7680 DATA 80000000000000000000000000000000 [1274]
,080
7690 DATA 0000000000000000000000000001D57D [2264]
,153
7700 DATA 555557F8000380000000000000000000 [1609]
,27C
7710 DATA 00000000000000000000000000000000 [2181]
,000
7720 DATA 0000001AA87AAAAF8000001000000000 [2340]
,37F
7730 DATA 00000000000000000000000000000000 [2181]
,000
7740 DATA 00000000000000000001D500D5558000 [1861]
,280
7750 DATA 00000000000000000000000000000000 [2181]
,000
7760 DATA 00000000000000000000000000000001 [2168]
,001
7770 DATA AA00AAA0000000000000000000000000 [2797]
,1FE
7780 DATA 00000000000000000000000000000000 [2181]
,000
7790 DATA 00000000001D40155540000000000000 [2217]
,17F
7800 DATA 00000000000000000000000000000000 [2181]
,000
7810 DATA 000000000000000000000001A802EAB8 [2414]
,24D
7820 DATA 00000000000000000000000000000000 [2181]
,000
7830 DATA 000000000000000000000000Q0000000 [2181]
,000
7840 DATA 0001D003576000000000000000000000 [2053]
,18B
7850 DATA 00000000000000000000000000000000 [2181]
,000
7860 DATA 0000000000000001A002AEC000000000 [2457]
,211
7870 DATA 00000000000000000000000000000000 [2181]
,000
7880 DATA 000000000000000000000000000 1C005 [2157]
,0C6
7890 DATA 55800000000000000000000000000000 [453]
,0D5
7900 DATA 00000000000000000000000000000000 [2181]
,000
7910 DATA 00000018003FE0000000000000000000 [1272]
,182
7920 DATA 00000000000000000000000000000000 [2181]
,000
7930 DATA 00000000000000000000180000000000 [1894]
,081
7940 DATA 00000000000000000000000000000000 [2181]
,000
7950 DATA 00000000000000000000000000000001 [2168]
,001
7960 DATA 80000000000000000000000000000000 [1274]
,080
7970 DATA 00000000000000000000000000000000 [2181]
,000
7980 DATA 00000000000180000000000000000000 [1870]
,081
7990 DATA 00000000000000000000000000000000 [2181]
,000
```

Listing HANSEAT

```
8000 DATA 00000000000000000000000180000000 [1862]
,081
8010 DATA 00000000000000000000000000000000 [2181]
,000
8020 DATA 00000000000000000000000000000000 [2181]
,000
8030 DATA 00018000000000000000000000000000 [1584]
,081
8040 DATA 00000000000000000000000000000000 [2181]
,000
8050 DATA 00000000000000018000000000000000 [1710]
,081
8060 DATA 00000000000000000000000000000000 [2181]
,000
8070 DATA 00000000000000000000000000018000 [2115]
,081
8080 DATA 00000000000000000000000000000000 [2181]
,000
8090 DATA 00000000000000000000000000000000 [2181]
,000
8100 DATA 00000018000000000000000000000000 [1774]
,081
8110 DATA 00000000000000000000000000000000 [2181]
,000
8120 DATA 00000000000000000000180000000000 [1894]
,081
8130 DATA 00000000000000000000000000000000 [2181]
,000
8140 DATA 00000000000000000000000000000001 [2168]
,001
8150 DATA 80000000000000000000000000000000 [1274]
,080
8160 DATA 00000000000000000000000000000000 [2181]
,000
8170 DATA 00000000000180000000000000000000 [1870]
,081
8180 DATA 00000000000000000000000000000000 [2181]
,000
8190 DATA 00000000000000000000000180000000 [1862]
,081
8200 DATA 00000000000000000000000000000000 [2181]
,000
8210 DATA 00000000000000000000000000000000 [2181]
,000
8220 DATA 00018000000000000000000000000000 [1584]
,081
8230 DATA 00000000000000000000000000000000 [2181]
,000
8240 DATA 00000000002000018CC3CCC3CDC3CFE0 [2282]
,63E
8250 DATA 0000000C00001E000C00100000000000 [1607]
,262
8260 DATA 03300000200000077001003000018C66 [2432]
,1D0
8270 DATA 6E666E666BA0000003E005E00335E003 [1911]
,596
8280 DATA E1800433E5E007F3C2E64211E00393E1 [2603]
,8A9
8290 DATA 83B3E2E18C6C6E66260C618003F00630 [3180]
,701
8300 DATA C770031630C631800634777006B46736 [1627]
,56F
8310 DATA 863310030631847637318FEFED63C7CF [1565]
,6C9
8320 DATA E18007E007E0C63001E610C7E1800666 [1981]
,7B0
8330 DATA 363006B666170631E003E7E18677E611 [1650]
,675
8340 DATA 8C6C6CE4660C618000000600C6300237 [1857]
,4D0
8350 DATA 30C6018003671630063666606C7743003 [2102]
,43D
8360 DATA 0601863606018C6E6CEC6E6E63800000 [1806]
,4DB
8370 DATA 03E0E2200636E0E3E1C00183E2200733 [2007]
,745
8380 DATA B20663D3E00703E1C3F3E20186666667 [3055]
,80B
8390 DATA CDC6630000000000400003E200400080 [1650]
,3DB
8400 DATA 00000000330000000000000600008000 [2101]
,0B9
8410 DATA 0001FFFFFFFFFFFFFFFFFFFFFFFFFFFF [2004]
,DF3
8420 DATA FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF [3602]
,FF0
8430 DATA FFFFFFFFFFFFFFFF0000000000000000 [2646]
,7F8
```

Listing HANSEAT

Ran an die Kisten

Fesselndes Denkspiel

Bei Let's work braucht man eine Portion Geschicklichkeit und logisches Denken. Das Spiel ist eine Umsetzung des MS-DOS-Spiels "Soko-Ban".

In dem Spiel muß die Spielfigur namens Pit in einem Neubauhaus mit verschiedenen Stockwerken, in dem die Bauarbeiter Unordnung hinterlassen haben, aufräumen, indem er die herumstehenden Kisten auf markierte Plätze schiebt. Das klingt zwar einfach, ist jedoch relativ schwer. Das Denkspiel ist eine interessante Abwechslung, wenn man Ballerspiele satt hat oder man keine Datenverarbeitung mehr sehen kann.

Pit muß in einem Neubauhaus mit verschiedenen Stockwerken (Levels), die in der Draufsicht dargestellt sind, Kisten auf gekennzeichnete Plätze schieben. Er hat aber nicht die Kraft, zwei Kisten auf einmal zu schieben, sondern kann immer nur eine bewegen. Auch kann er die Kisten nur schieben, nicht ziehen oder tragen. In manchen Stockwerken breitet sich zusätzlich noch eine Flüssigkeit (Waber) aus, mit der Pit nicht in Berührung kommen darf. Die Flüssigkeit kann sich nur auf freien Stellen ausbreiten, also nicht auf Mauern, gekennzeichneten Stellen oder Kisten. Wenn das Waber einmal eingesperrt ist, breitet es sich nicht mehr weiter aus, auch wenn es wieder die Möglichkeit dazu hätte, zum Beispiel weil eine Kiste, die das Ausbreiten verhindert hatte, weggeschoben wurde. Wichtig ist, daß nur alle gekennzeichneten Plätze mit Kisten bedeckt sein, nicht aber alle Kisten auf markierten Plätzen stehen müssen! Das ist vor allem in den Levels wichtig, in denen es verschieden viele Kisten und gekennzeichnete Plätze gibt. Wenn dann alle gekennzeichneten Plätze mit Kisten bedeckt sind, kommt man in den nächsten Level. Wenn man den letzten Level geschafft hat, kommt man wieder in den ersten. Es gibt keine Punktewertung oder Highscoretabelle, es geht nur darum, den nächsten Level zu erreichen. Das Programm besteht aus vier Listings:

1. LETSWORK.BAS: das Hauptprogramm
2. LWBINLAD.DAT: Datalader, der den Maschinenspracheteil "LETSWORK. BIN" des Spiels erzeugt und speichert
3. CLBINLAD.DAT: zweiter Datalader, der COMPLEVE.BIN mit 30 komprimierten Levels erzeugt und speichert
4. ENTCOMP.BAS: Programm, das aus den komprimierten Levels normale erzeugt und unter "LEVEL.LET" speichert

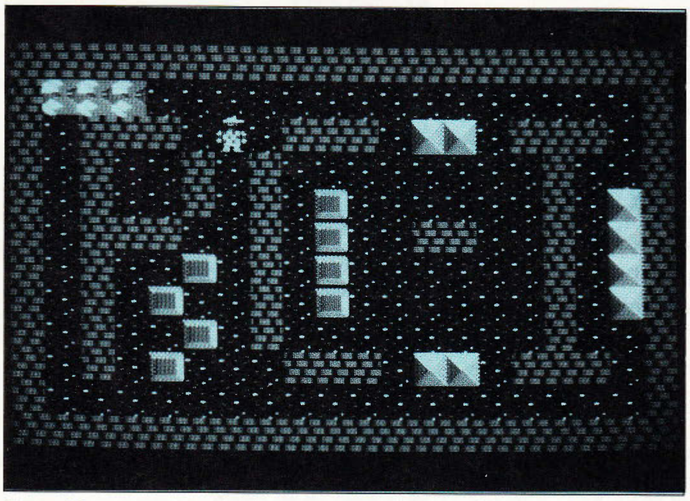
Kassettenbetrieb

Bei Kassettenbetrieb muß man darauf achten, daß zuerst das BASIC-Programm auf der Kassette gespeichert ist, dann der Maschinenspracheteil, der vom BASIC-Programm automatisch nachgeladen wird, und danach dann die Leveldatei(en). Bei Kassettenbetrieb ist folgendes Vorgehen beim Abtippen empfehlenswert:

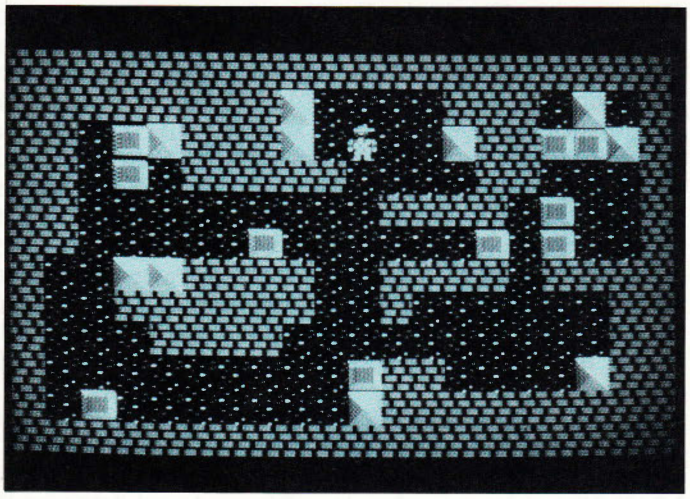
1. Abtippen des BASIC-Programms und Speichern auf Kassette A unter dem Namen "LETSWORK.BAS"
2. Abtippen des Dataladers "LWBINLAD.DAT" und Speichern auf Kassette B mit dem Namen "LWBINLAD.DAT"
3. An das Ende von "LETSWORK.BAS" gespulte Kassette A einlegen und im Speicher befindliches "LWBINLAD.DAT" starten ("LETSWORK.BIN" wird dann automatisch hinter "LETSWORK.BAS" gespeichert.).
4. Abtippen des Dataladers "CLBINLAD.DAT" und auf Kassette B speichern, noch nicht starten!
5. "ENTCOMP.BAS" abtippen und auf Kassette B hinter "CLBINLAD.DAT" speichern, ebenfalls noch nicht starten!
6. "CLBINLAD.DAT" laden, Kassette B an das Ende von "ENTCOMP.BAS" spulen und im Speicher befindliches Programm "CLBINLAD.DAT" mit RUN starten. Es wird nun die Binärdatei "COMPLEVE.BIN" erzeugt und hinter "ENTCOMP.BAS" gespeichert
7. "ENTCOMP.BAS" starten. Es wird "COMPLEVE.BIN" nachgeladen. Auf Aufforderung die an das Ende von "LETSWORK.BIN" gespulte Kassette A einlegen ("LEVEL.LET" wird dann automatisch hinter "LETSWORK.BIN" gespeichert.). Dies sieht viel komplizierter aus, als es ist. Selbst Anfänger müßten mit dieser Anleitung zurechtkommen.

Diskettenbenutzer

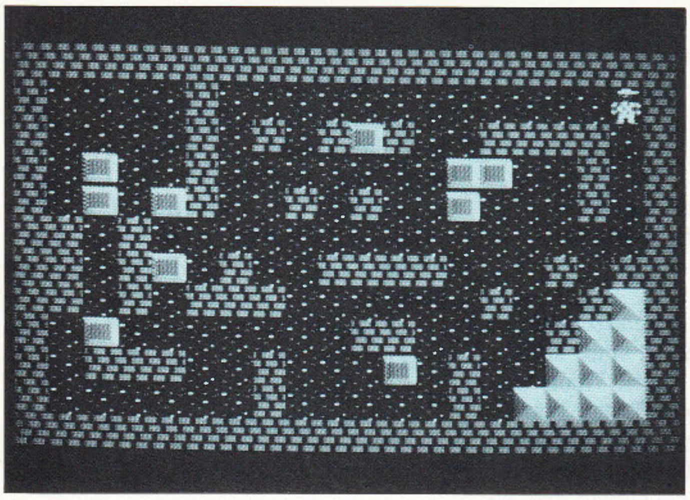
Für Diskettenbenutzer ist es nicht ganz so kompliziert wie beim Kassettenbetrieb. Hier muß man nur beachten, daß man, bevor man "ENTCOMP.BAS" startet, "CLBINLAD.DAT" abgetippt und gestartet hat. Wenn man mit dem Abtippen fertig ist und getestet hat, ob das Spiel läuft, braucht man nur noch folgende Dateien auf der Diskette bzw. Kassette haben:



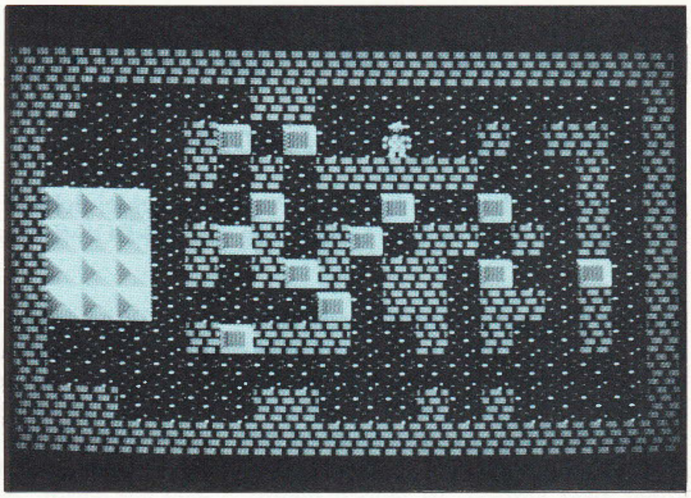
Bis zu 80 verschiedene Level können zusätzlich zum Editor im Spiel verwendet werden.



Im ersten Level....



... doch das zweite folgt zu gleich



Unlösbar erscheinen manche Spielstufen, aber mit einigem Geschick läßt sich die Lage klären

1. LETSWORK.BAS
2. LETSWORK.BIN
3. LEVEL.LET

Die anderen Dateien können dann gelöscht werden. Beim Abtippen kann man sämtliche REM-Zeilen weglassen.

Spielstart

Gestartet wird das Spiel mit: *run"letswork"*. Der Speicher sollte möglichst leer sein, deshalb ist ein Reset des Computers vor dem Start des Spiels günstig. Es erscheint dann das Titelbild und das Menü mit den Funktionen Game, Joystick, Tastatur, Load, Save, Editor und BASIC. Um eine Funktion anzuwählen, muß der erste Buchstabe der Funktion, der auch in einer anderen Farbe dargestellt ist, eingegeben werden.

GAME: Spielstart. Hierzu muß sich mindestens ein Level im Speicher befinden, anderenfalls passiert nichts. Dann wird das Level mit den Cursortasten auf (Level erhöhen) und ab (Level erniedrigen) oder dem Joystick (oben und unten) ausgewählt. Wenn man sich für ein Level entschieden hat, drückt man die RETURN- bzw. die große ENTER-Taste (Tastennummer 18) oder den Feuerknopf des Joysticks. Jetzt erscheint das Level, und man kann das Männchen, je nach Einstellung mit den Cursortasten oder dem Joystick, bewegen. Das Spiel kann jederzeit mit der Escape-Taste beendet werden. Das ist oft nötig, wenn man sich in eine Situation manövriert hat, in der man das Level nicht mehr schaffen kann.

JOYSTICK: Dieser Menüpunkt schaltet von Tastatursteuerung auf Joysticksteuerung um. Das zeigt sich nur während des Spieles, nicht aber bei der Levelauswahl oder im Editor.

TASTATUR: Hier wird auf Tastatursteuerung umgeschaltet. Dies wirkt sich wiederum nur auf das Spiel aus.

LOAD: Hiermit kann man Level von Diskette (oder Kassette) laden. Die Leveldateien haben die Extension ".LET". Zuerst muß der Name der zu ladenden Leveldatei eingegeben werden. Eine Leereingabe (RETURN) bewirkt einen Rücksprung ins Hauptmenü. Danach wird mit den Cursortasten oder dem Joystick und RETURN oder Feuer das Level ausgewählt, an den die neuen Level hingeladen werden sollen.

SAVE: Mit dieser Funktion kann man Levels aus dem Speicher auf Diskette abspeichern. Zuerst den Namen eingegeben, RETURN bewirkt auch hier einen Rücksprung zum Hauptmenü. Als nächstes wählt man, wie auch bei der LOAD-Funktion, das Level aus, ab dem gespeichert werden soll, danach das Level, bis zu dem gespeichert werden soll.

EDITOR: Diese Funktion ist zum Ändern und Neuerstellen von Bildern. Man wählt das zu bearbeitende Level aus. Dann kann man mit den Cursortasten ein Kreuz über den Bildschirm bewegen. Mit Druck auf die Taste "M" erscheint eine Mauer an der Stelle des Kreuzes. "K" setzt eine Kiste, "N" definiert den Untergrund als normal, "G" als gekennzeichnet. Wenn bei den Funktionen N und G eine Kiste an der Position des Kreuzes steht, wird diese nicht gelöscht! Der markierte bzw. normale Untergrund wird dann nicht gezeichnet und ist nur sichtbar, falls man die Kiste wieder löschen sollte. Dies läßt sich mit der Taste "L" machen. Übrig bleibt nur noch die Taste "C", die den ganzen Bildschirm löscht. Hat man das Bild entworfen, drückt man die RETURN-Taste. Nun muß festgelegt werden, an welcher Stelle Pit am Anfang stehen soll. Wurde diese Stelle schon einmal festgelegt, kann sie mit RETURN übernommen werden. Anderenfalls bewegt man wieder das Kreuz über den Bildschirm und setzt mit Druck auf die Leertaste Pit an die momentane Position. Hat man sich für eine Position entschieden, drückt man RETURN. Nun wählt man die Position des Wabers aus, falls Waber erwünscht ist. Wenn nicht, kann man bei neuen Levels einfach RETURN drücken. Bei alten Levels kann vorhandenes Waber mit der DEL-Taste gelöscht werden. Um Waber an die momentane Kreuzposition zu setzen, drückt man die Leertaste. Wenn man zufrieden ist, kommt man mit RETURN wieder ins Hauptmenü zurück. Es ist möglich, etwa 80 Levels gleichzeitig im Speicher zu haben. Man sollte beim Entwerfen von Levels darauf achten, daß um jedes Bild ein Rahmen, bestehend aus Mauerstücken, ist.

BASIC: Das Programm wird hiermit beendet und kehrt zum BASIC-Direktmodus zurück. Ein Neustart des Programm ist zwar möglich, sollte jedoch vermieden werden. Einige der abgedruckten Levels sind wohl kaum auf Anhieb lösbar, es sind jedoch keine unmöglichen Levels dabei.

(Stefan Diem/cd)

Variablenliste
VAL(a$) basis : Basisadresse der Levels
i : Zählvariable
iflg : Flag, ob Mann/Waber schon einmal gesetzt worden ist (Editor)
lev : Level, das gespielt wird, an den Level geladen werden sollen, bis zu dem Level gespeichert werden sollen, der editiert werden
levl : Level, ab dem gespeichert werden soll
maxlev: momentane Anzahl der Levels
neuflg: Flag, ob Level, das editiert werden soll, neu ist
s : Zufallzahl, die den nächsten Dreiklangton für die Titelmelodie auswählt (s: 0-4)
x : Zählvariable / x-Position des Kreuzes im Editor
xa : Zählvariable
xe : x-Position des Kreuzes im Editor
xm : Zählvariable / x-Position des Mannes im Editor
xw : x-Position des Wabers
y : Zählvariable / y-Position des Kreuzes im Editor
ya : Zählvariable
ye : y-Position des Kreuzes im Editor
ym : Zählvariable / y-Position des Mannes im Editor
yw : y-Position des Wabers
z : Zählvariable
f(m,n): Wert der Bildposition m,n des im Editor gerade behandelten Bildes; m: 0-19 (x-Position), n: 0-11 (y-Position)
s(m,n): m-ter Dreiklangton des n-ten Dreiklangs (Titelmelodie); m: 1-3, n: 0-4
so(m) : Adresse der Daten des m-ten Shapes; m: 0,1,2,128,129
a$: Zeichen von der Tastatur
n$: Name des zu ladenden / speichernden Programms

für 464-664-6128

```
10 REM                                              [272]
20 REM Letswork Basic Programm                      [1606]
30 REM                                              [272]
40 IF PEEK(&9E77)<>33 THEN MEMORY &3FFF:LO [3897]
AD"letswork.bin"
50 POKE &9838,0:POKE &98E6,2:POKE &9997,8: [4507]
POKE &9A45,1:DEFINT a-z
60 ENT-2,3,3,3,3,-3,3:ENV 2,7,1,1             [1931]
70 s(1,0)=239:s(1,1)=213:s(1,2)=190:s(1,3) [10107]
=179:s(1,4)=159:s(2,0)=190:s(2,1)=169:s(2,
2)=159:s(2,3)=142:s(2,4)=127:s(3,0)=159:s(
3,1)=142:s(3,2)=127:s(3,3)=119:s(3,4)=106:
DIM f(19,11),so(129)
80 so(0)=&A480:so(2)=&A200:so(1)=&A240:so( [2679]
128)=&A4C0:so(129)=&A240
90 'Titelbildaufbau                           [1256]
100 WHILE INKEY$<>"":WEND:MODE 1:BORDER 0: [6796]
INK 0,0:INK 1,0:INK 2,0:INK 3,0:PAPER 0:CA
LL &BD19
110 LOCATE 29,15:PEN 3:PRINT CHR$(164)" by [5407]
":LOCATE 27,17:PRINT"Stelo 1988"
120 LOCATE 4,12:PEN 1:PRINT"G";:PEN 2:PRIN [1750]
T"ame"
130 LOCATE 6,14:PEN 1:PRINT"J";:IF PEEK(&9 [3891]
838)=0 THEN PEN 2 ELSE PEN 3
140 PRINT"oystick"                            [1035]
150 LOCATE 8,16:PEN 1:PRINT"T";:IF PEEK(&9 [4580]
838) THEN PEN 2 ELSE PEN 3
160 PRINT"astatur"                            [1299]
170 LOCATE 10,18:PEN 1:PRINT"E";:PEN 2:PRI [3295]
NT"ditor"
180 LOCATE 12,20:PEN 1:PRINT"L";:PEN 2:PRI [1717]
NT"oad"
190 LOCATE 14,22:PEN 1:PRINT"S";:PEN 2:PRI [3344]
NT"ave"
200 LOCATE 16,24:PEN 1:PRINT"B";:PEN 2:PRI [2727]
NT"asic"
210 INK 1,26:INK 2,15:INK 3,2:SOUND 1,s(1, [8385]
0),50,4,2,2:SOUND 2,s(2,0),50,4,2,2:SOUND
4,s(3,0),50,4,2,2:SOUND 1,s(1,1),50,4,2,2:
SOUND 2,s(2,1),50,4,2,2:SOUND 4,s(3,1),50,
4,2,2
220 SOUND 1,s(1,2),50,4,2,2:SOUND 2,s(2,2) [5224]
,50,4,2,2:SOUND 4,s(3,2),50,4,2,2:CALL &9E
77
230 WHILE 1:s=RND*4:SOUND 1,s(1,s),50,4,2, [4992]
2:SOUND 2,s(2,s),50,4,2,2:SOUND 4,s(3,s),5
0,4,2,2
240 a$=LOWER$(INKEY$):IF a$=""THEN 320        [1486]
250 IF a$="g"THEN GOSUB 1100:GOTO 340         [1613]
260 IF a$="t"THEN 440                         [886]
270 IF a$="j"THEN 460                         [1310]
280 IF a$="l"THEN GOSUB 1100:GOTO 480         [1980]
290 IF a$="s"THEN GOSUB 1100:GOTO 540         [2564]
300 IF a$="e"THEN GOSUB 1100:GOTO 720         [1944]
310 IF a$="b"THEN MODE 1:PEN 1:GOSUB 1100: [2123]
END
320 WEND                                      [390]
330 'Spiel                                    [141]
340 IF maxlev=0 THEN 230                      [464]
350 IF maxlev=1 THEN lev=1:GOTO 370           [1564]
360 PEN 2:LOCATE 20,21:PRINT"Level :";:max [5168]
lev=maxlev-1:GOSUB 680:maxlev=maxlev+1
370 basis=&3F0B+lev*246:FOR i=&9696 TO &97 [9342]
85:POKE i,PEEK(basis):basis=basis+1:NEXT:P
OKE &9786,(PEEK(basis)-1)/2:basis=basis+1:
POKE &9787,(PEEK(basis)-1)/2:basis=basis+1
:POKE &9788,0
380 IF PEEK(basis)<>255 THEN POKE &9788,1: [9616]
POKE &9DF0,(PEEK(basis)-1)/2:basis=basis+1
:POKE &9DF1,(PEEK(basis)-1)/2:POKE &9696+(
PEEK(basis-1)-1)/2+(PEEK(basis)-1)/2*20,5
390 FOR i=&9DF2 TO &9DF5:POKE i,0:NEXT:POK [2474]
E &9789,0
400 CALL &BB48:CALL &978B:CALL &BB45:GOSUB [9018]
 1100:IF PEEK(&9789)=123 THEN FOR i=1500 T
O 2000 STEP 70:SOUND 1,i,20:NEXT:WHILE SQ(
1)<>4:WEND:GOTO 100
410 FOR i=200 TO 100 STEP -10:SOUND 1,i,5: [6389]
NEXT:lev=lev+1:IF lev>maxlev THEN lev=1
420 GOTO 370                                  [466]
430 'Tastatur-Steuerung                       [1293]
```

Listing LETSWORK

```
440 POKE &9838,0:POKE &98E6,2:POKE &9997,8 [8933]
:POKE &9A45,1:LOCATE 6,14:PEN 1:PRINT"J";:
PEN 2:PRINT"oystick":LOCATE 8,16:PEN 1:PRI
NT"T";:PEN 3:PRINT"astatur":GOTO 230
450 'Joystick-Steuerung                       [1869]
460 POKE &9838,72:POKE &98E6,73:POKE &9997 [9854]
,74:POKE &9A45,75:LOCATE 8,16:PEN 1:PRINT"
T";:PEN 2:PRINT"astatur":LOCATE 6,14:PEN 1
:PRINT"J";:PEN 3:PRINT"oystick":GOTO 230
470 'Level laden                              [1862]
480 GOSUB 610:IF n$=""THEN LOCATE 20,21:PR [3635]
INT SPACE$(19):GOTO 230
490 IF maxlev=0 THEN LOAD n$+".let",&4000: [2453]
GOTO 520
500 LOCATE 17,21:PRINT"an Level :";:GOSUB  [3718]
1120:GOSUB 680
510 LOAD n$+".let",&3F0A+lev*246              [1837]
520 i=&4000+246*maxlev:z=0:WHILE PEEK(i)=1 [6907]
:i=i+246:z=z+1:WEND:maxlev=maxlev+z:LOCATE
 17,21:PRINT SPACE$(18);:GOTO 230
530 'Level speichern                          [2030]
540 IF maxlev=0 THEN 230                      [464]
550 GOSUB 610:IF n$=""THEN LOCATE 20,21:PR [3635]
INT SPACE$(19):GOTO 230
560 IF maxlev=1 THEN levl=1:lev=1:maxlev=m [3232]
axlev-1:GOTO 590
570 LOCATE 16,21:maxlev=maxlev-1:PRINT"von [4718]
 Level :";:GOSUB 1120:GOSUB 680
580 levl=lev:LOCATE 16,21:PEN 2:PRINT"bis  [5505]
Level :";:GOSUB 1120:GOSUB 680:IF levl>lev
 THEN PRINT CHR$(7);:GOTO 570
590 maxlev=maxlev+1:SAVE n$+".let",b,&3F0A [5943]
+246*levl,(lev-levl+1)*246:LOCATE 16,21:PR
INT SPACE$(20);:GOTO 230
600 'Nameneingabe                             [797]
610 LOCATE 20,21:PEN 1:PRINT"Name :";:PEN  [5552]
2:PRINT" ........";:LOCATE 27,21:n$="":FOR
 i=1 TO 8
620 a$=LOWER$(INKEY$):IF a$=""THEN 620        [1538]
630 IF a$=CHR$(13)THEN RETURN                 [1139]
640 IF a$=CHR$(127)AND LEN(n$)<>0 THEN n$= [4577]
LEFT$(n$,LEN(n$)-1):PRINT CHR$(8)"."CHR$(8
);:i=i-1:GOTO 620
650 IF a$<"0"OR a$>"z"THEN 620                [1452]
660 PRINT a$;:n$=n$+a$:NEXT:RETURN            [1372]
670 'Leveleingabe                             [2251]
680 PEN 3:PRINT" 001    ":lev=1:WHILE INKE [12808]
Y(18)AND JOY(0)<>16:IF(NOT INKEY(0)OR JOY(
0)=1)AND lev<maxlev+1 THEN lev=lev+1:LOCAT
E 28,21:PRINT RIGHT$("00"+RIGHT$(STR$(lev)
,LEN(STR$(lev))-1),3)
690 IF(NOT INKEY(2)OR JOY(0)=2)AND lev>1 T [8666]
HEN lev=lev-1:LOCATE 28,21:PRINT RIGHT$("0
0"+RIGHT$(STR$(lev),LEN(STR$(lev))-1),3)
700 FOR i=1 TO 100:NEXT:WEND:SOUND 1,50,5: [2339]
RETURN
710 'Editor                                   [630]
720 IF maxlev<>0 THEN LOCATE 20,21:PRINT"L [3240]
evel :";:GOSUB 680 ELSE lev=1
730 neuflg=0:IF lev>maxlev THEN neuflg=-1     [2610]
740 PEN 1:basis=&3F0A+lev*246+1:FOR y=0 TO [8367]
 11:FOR x=0 TO 19:f(x,y)=PEEK(basis):basis
=basis+1:CALL &9F00,x,y,so(f(x,y)):NEXT:NE
XT:xm=0:ym=0:xw=0:yw=0
750 x=1:y=1:xe=0:ye=0:GOSUB 1130              [2216]
760 GOSUB 1000                                [851]
770 IF NOT INKEY(18)THEN SOUND 1,50,5:CALL [10065]
 &9F00,xe,ye,so(f(xe,ye)):basis=&3F0A+lev*
246:POKE basis,1:basis=basis+1:FOR ym=0 TO
 11:FOR xm=0 TO 19:POKE basis,f(xm,ym):bas
is=basis+1:NEXT:NEXT:GOTO 850
780 IF NOT INKEY(38)THEN f(xe,ye)=2:CALL & [5313]
9F00,xe,ye,&A200:GOTO 760
790 IF NOT INKEY(37)THEN f(xe,ye)=f(xe,ye) [3585]
AND 128 OR 1:CALL &9F00,xe,ye,&A240:GOTO 7
60
800 IF NOT INKEY(46)THEN f(xe,ye)=f(xe,ye) [5921]
AND 1:CALL &9F00,xe,ye,so(f(xe,ye)):GOTO 7
60
810 IF NOT INKEY(52)THEN f(xe,ye)=f(xe,ye) [6288]
AND 1 OR 128:CALL &9F00,xe,ye,so(f(xe,ye))
:GOTO 760
820 IF NOT INKEY(62)THEN FOR xa=0 TO 19:FO [8580]
R ya=0 TO 11:f(xa,ya)=0:CALL &9F00,xa,ya,&
A480:NEXT:NEXT:GOSUB 1130:GOSUB 1120:GOTO
760
```

Listing LETSWORK

```
830 IF NOT INKEY(36)THEN IF f(xe,ye)<>1 AN [7427]
D f(xe,ye)<>129 THEN 760 ELSE f(xe,ye)=f(x
e,ye)AND 128:CALL &9F00,xe,ye,so(f(xe,ye))
:GOTO 760
840 GOTO 760                               [413]
850 x=1:y=1:xe=0:ye=0:xm=0:ym=0:GOSUB 1130 [8244]
:iflg=0:IF PEEK(basis)<>0 THEN xm=PEEK(bas
is):ym=PEEK(basis+1):CALL &9F00,(xm-1)/2,(
ym-1)/2,&A2C0:iflg=-1
860 GOSUB 1000                             [851]
870 IF NOT INKEY(47)THEN IF iflg THEN CALL [13307]
 &9F00,(xm-1)/2,(ym-1)/2,so(f((xm-1)/2,(ym
-1)/2)):CALL &9F00,xe,ye,&A2C0:xm=x:ym=y:G
OTO 860 ELSE iflg=-1:CALL &9F00,xe,ye,&A2C
0:xm=x:ym=y:GOTO 860
880 IF INKEY(18)OR NOT iflg THEN 860       [1095]
890 IF xm<>x OR ym<>y THEN CALL &9F00,xe,y [2914]
e,so(f(xe,ye))
900 f((xm-1)/2,(ym-1)/2)=0:POKE &3F0A+lev* [9530]
246+1+((ym-1)/2)*20+(xm-1)/2,0:POKE basis,
xm:basis=basis+1:POKE basis,ym:basis=basis
+1:x=1:y=1:xe=0:ye=0:xw=0:yw=0:GOSUB 1130:
iflg=0
910 GOSUB 1120:IF(PEEK(basis)<>0 AND PEEK( [8472]
basis)<>255)THEN xw=PEEK(basis):yw=PEEK(ba
sis+1):CALL &9F00,(xw-1)/2,(yw-1)/2,&A280:
iflg=-1
920 GOSUB 1000                             [851]
930 IF NOT INKEY(47)AND(x<>xm OR y<>ym)THE [11827]
N IF iflg THEN CALL &9F00,(xw-1)/2,(yw-1)/
2,so(f((xw-1)/2,(yw-1)/2)):CALL &9F00,xe,y
e,&A280:xw=x:yw=y:GOTO 920 ELSE iflg=-1:CA
LL &9F00,xe,ye,&A280:xw=x:yw=y:GOTO 920
940 IF NOT INKEY(79)AND iflg THEN iflg=0:C [8870]
ALL &9F00,(xw-1)/2,(yw-1)/2,so(f((xw-1)/2,
(yw-1)/2)):xw=0:yw=0:GOSUB 1130
950 IF INKEY(18)THEN 920                   [1248]
960 IF NOT iflg THEN POKE basis,255:basis= [8843]
basis+1:POKE basis,255:basis=basis+1 ELSE
POKE basis,xw:basis=basis+1:POKE basis,yw:
basis=basis+1
970 IF neuflg THEN maxlev=maxlev+1         [1506]
980 GOTO 100                               [417]
990 'Kreuzsteuerung                        [1573]
1000 IF NOT INKEY(0)AND y<>1 THEN GOSUB 10 [4162]
60:y=y-2:ye=ye-1:GOSUB 1130
1010 IF NOT INKEY(2)AND y<>23 THEN GOSUB 1 [3643]
060:y=y+2:ye=ye+1:GOSUB 1130
1020 IF NOT INKEY(8)AND x<>1 THEN GOSUB 10 [3825]
60:x=x-2:xe=xe-1:GOSUB 1130
1030 IF NOT INKEY(1)AND x<>39 THEN GOSUB 1 [3401]
060:x=x+2:xe=xe+1:GOSUB 1130
1040 RETURN                                [555]
1050 'Untergrund zurueckschreiben          [2564]
1060 IF x=xm AND y=ym THEN CALL &9F00,xe,y [3414]
e,&A2C0:RETURN
1070 IF x=xw AND y=yw THEN CALL &9F00,xe,y [2935]
e,&A280:RETURN
1080 CALL &9F00,xe,ye,so(f(xe,ye)):RETURN  [3523]
1090 'Kontrollsound                        [929]
1100 SOUND 135,50,5:RETURN                 [1488]
1110 'Verzoegerungsschleife+Tastaturpuffer [2279]
 leeren
1120 FOR i=1 TO 150:NEXT:WHILE INKEY$<>"": [1646]
WEND:RETURN
1130 LOCATE x,y:PRINT"\/";:LOCATE x,y+1:PR [4549]
INT"/\";:RETURN
```

```
100 '*********************************** [1460]
101 '* LWBINLAD - DATA-Lader von 'CPC' * [2108]
102 '*********************************** [1460]
103 '                                     [117]
104 DATA 9696,00,00,00,00,00,00,00,00,00,0 [830]
0,00,00,00,00,00,00,0000
105 DATA 96A6,00,00,00,00,00,00,00,00,00,0 [3033]
0,00,00,00,00,00,00,0000
106 DATA 96B6,00,00,00,00,00,00,00,00,00,0 [2249]
0,00,00,00,00,00,00,0000
107 DATA 96C6,00,00,00,00,00,00,00,00,00,0 [3166]
0,00,00,00,00,00,00,0000
108 DATA 96D6,00,00,00,00,00,00,00,00,00,0 [2124]
0,00,00,00,00,00,00,0000
109 DATA 96E6,00,00,00,00,00,00,00,00,00,0 [1984]
0,00,00,00,00,00,00,0000
110 DATA 96F6,00,00,00,00,00,00,00,00,00,0 [3147]
0,00,00,00,00,00,00,0000
```

Listing LETSWORK

```
111 DATA 9706,00,00,00,00,00,00,00,00,00,0 [2755]
0,00,00,00,00,00,00,0000
112 DATA 9716,00,00,00,00,00,00,00,00,00,0 [2029]
0,00,00,00,00,00,00,0000
113 DATA 9726,00,00,00,00,00,00,00,00,00,0 [2118]
0,00,00,00,00,00,00,0000
114 DATA 9736,00,00,00,00,00,00,00,00,00,0 [2794]
0,00,00,00,00,00,00,0000
115 DATA 9746,00,00,00,00,00,00,00,00,00,0 [3093]
0,00,00,00,00,00,00,0000
116 DATA 9756,00,00,00,00,00,00,00,00,00,0 [3124]
0,00,00,00,00,00,00,0000
117 DATA 9766,00,00,00,00,00,00,00,00,00,0 [2822]
0,00,00,00,00,00,00,0000
118 DATA 9776,00,00,00,00,00,00,00,00,00,0 [1243]
0,00,00,00,00,00,00,0000
119 DATA 9786,00,00,00,00,00,21,F6,9D,3E,0 [1808]
1,CD,BC,BC,21,FD,9D,FC22
120 DATA 9796,3E,01,CD,BF,BC,DD,21,2B,9B,0 [3176]
1,00,00,CD,18,9B,C5,A730
121 DATA 97A6,B7,20,06,21,80,A4,C3,04,9B,F [2383]
E,02,20,06,21,00,A2,3200
122 DATA 97B6,C3,04,9B,FE,80,20,06,21,C0,A [3122]
4,C3,04,9B,21,40,A2,F826
123 DATA 97C6,C3,04,9B,C1,0C,79,FE,14,20,D [3016]
2,0E,00,04,78,FE,0C,B90F
124 DATA 97D6,20,CA,ED,4B,86,97,DD,70,02,D [3468]
D,71,04,11,C0,A2,CD,0C1D
125 DATA 97E6,00,9F,3A,88,97,B7,28,30,3A,F [2847]
0,9D,32,26,9D,3A,F1,366F
126 DATA 97F6,9D,32,8B,9D,97,32,F2,9D,3C,3 [3621]
2,F3,9D,ED,4B,F0,9D,BB56
127 DATA 9806,DD,71,04,DD,70,02,11,80,A2,C [1775]
D,00,9F,01,0F,00,11,D1E9
128 DATA 9816,14,00,21,5F,9E,CD,E9,BC,21,9 [2211]
6,96,7E,FE,80,28,09,5F84
129 DATA 9826,23,7D,FE,86,20,F5,C3,6B,9E,3 [5003]
A,89,97,FE,7B,CA,6B,0729
130 DATA 9836,9E,3E,48,CD,1E,BB,CA,E5,98,3 [3551]
A,88,97,B7,28,14,ED,626B
131 DATA 9846,4B,86,97,05,CD,18,9B,FE,05,2 [2596]
0,08,3E,7B,32,89,97,7AE4
132 DATA 9856,C3,6B,9E,3A,87,97,3D,CA,47,9 [2038]
E,47,3A,86,97,4F,CD,FD12
133 DATA 9866,18,9B,32,89,97,FE,02,CA,47,9 [2660]
E,E6,7F,28,06,78,FE,8045
134 DATA 9876,02,DA,47,9E,05,CD,18,9B,32,8 [2358]
A,97,E6,7F,28,08,3A,1347
135 DATA 9886,89,97,E6,7F,C2,47,9E,3A,89,9 [2882]
7,E6,7F,28,08,3A,8A,0192
136 DATA 9896,97,FE,02,CA,47,9E,21,86,97,4 [2592]
E,23,46,DD,71,06,DD,3B85
137 DATA 98A6,70,04,CD,18,9B,E6,80,DD,77,0 [3540]
2,3A,89,97,E6,7F,DD,8EF4
138 DATA 98B6,77,00,C4,54,9E,CD,83,A0,3A,8 [2992]
7,97,3D,32,87,97,47,0D35
139 DATA 98C6,3A,86,97,4F,CD,18,9B,32,89,9 [2429]
7,E6,7F,CA,1E,98,3A,E17F
140 DATA 98D6,89,97,E6,FE,77,05,CD,18,9B,F [3620]
6,01,77,C3,1E,98,3E,3E4E
141 DATA 98E6,49,CD,1E,BB,CA,96,99,3A,88,9 [4073]
7,B7,28,14,ED,4B,86,C6BE
142 DATA 98F6,97,04,CD,18,9B,FE,05,20,08,3 [3241]
E,7B,32,89,97,C3,6B,AA6C
143 DATA 9906,9E,3A,87,97,FE,0A,CA,47,9E,3 [2749]
C,47,3A,86,97,4F,CD,5621
144 DATA 9916,18,9B,32,89,97,FE,02,CA,47,9 [3260]
E,E6,7F,CA,2B,99,78,86A5
145 DATA 9926,FE,0A,D2,47,9E,04,CD,18,9B,3 [3086]
2,8A,97,E6,7F,28,08,8DC2
146 DATA 9936,3A,89,97,E6,7F,C2,47,9E,3A,8 [4065]
9,97,E6,7F,28,08,3A,0091
147 DATA 9946,8A,97,FE,02,CA,47,9E,21,86,9 [1818]
7,4E,23,46,DD,71,06,1DE9
148 DATA 9956,DD,70,04,CD,18,9B,E6,80,DD,7 [2371]
7,02,3A,89,97,E6,7F,477A
149 DATA 9966,DD,77,00,C4,54,9E,CD,1A,A1,3 [2984]
A,87,97,3C,32,87,97,1F5F
150 DATA 9976,47,3A,86,97,4F,CD,18,9B,32,8 [3475]
9,97,E6,7F,CA,1E,98,7081
151 DATA 9986,3A,89,97,E6,FE,77,04,CD,18,9 [2265]
B,F6,01,77,C3,1E,98,1D25
152 DATA 9996,3E,4A,CD,1E,BB,CA,44,9A,3A,8 [2372]
8,97,B7,28,14,ED,4B,0402
153 DATA 99A6,86,97,0D,CD,18,9B,FE,05,20,0 [3622]
8,3E,7B,32,89,97,C3,F541
154 DATA 99B6,6B,9E,3A,86,97,3D,CA,47,9E,4 [2758]
F,3A,87,97,47,CD,18,E8CE
```

Listing LETSWORK

```
155 DATA 99C6,9B,32,89,97,FE,02,CA,47,9E,E [2994]
6,7F,28,06,79,FE,02,0095
156 DATA 99D6,DA,47,9E,0D,CD,18,9B,32,8A,9 [3181]
7,E6,7F,28,08,3A,89,A605
157 DATA 99E6,97,E6,7F,C2,47,9E,3A,89,97,E [3193]
6,7F,28,08,3A,8A,97,033A
158 DATA 99F6,FE,02,CA,47,9E,21,86,97,4E,2 [4394]
3,46,DD,71,06,DD,70,77ED
159 DATA 9A06,04,CD,18,9B,E6,80,DD,77,02,3 [3549]
A,89,97,E6,7F,DD,77,1DEE
160 DATA 9A16,00,C4,54,9E,CD,E7,9F,3A,86,9 [2941]
7,3D,32,86,97,4F,3A,F4BE
161 DATA 9A26,87,97,47,CD,18,9B,32,89,97,E [2466]
6,7F,CA,1E,98,3A,89,424D
162 DATA 9A36,97,E6,FE,77,0D,CD,18,9B,F6,0 [2468]
1,77,C3,1E,98,3E,4B,3C5E
163 DATA 9A46,CD,1E,BB,CA,F4,9A,3A,88,97,B [3646]
7,28,14,ED,4B,86,97,91A0
164 DATA 9A56,0C,CD,18,9B,FE,05,20,08,3E,7 [2541]
B,32,89,97,C3,6B,9E,54D4
165 DATA 9A66,3A,86,97,FE,12,CA,47,9E,3C,4 [2902]
F,3A,87,97,47,CD,18,2EF4
166 DATA 9A76,9B,32,89,97,FE,02,CA,47,9E,E [2697]
6,7F,28,06,79,FE,12,0085
167 DATA 9A86,D2,47,9E,0C,CD,18,9B,32,8A,9 [3573]
7,E6,7F,28,08,3A,89,B601
168 DATA 9A96,97,E6,7F,C2,47,9E,3A,89,97,E [4049]
6,7F,28,08,3A,8A,97,033A
169 DATA 9AA6,FE,02,CA,47,9E,21,86,97,4E,2 [3488]
3,46,DD,71,06,DD,70,77ED
170 DATA 9AB6,04,CD,18,9B,E6,80,DD,77,02,3 [3228]
A,89,97,E6,7F,DD,77,1DEE
171 DATA 9AC6,00,C4,54,9E,CD,53,9F,3A,86,9 [4648]
7,3C,32,86,97,4F,3A,249C
172 DATA 9AD6,87,97,47,CD,18,9B,32,89,97,E [2935]
6,7F,CA,1E,98,3A,89,424D
173 DATA 9AE6,97,E6,FE,77,0C,CD,18,9B,F6,0 [2895]
1,77,C3,1E,98,3E,42,3457
174 DATA 9AF6,CD,1E,BB,CA,1E,98,3E,7B,32,8 [3449]
9,97,C3,6B,9E,DD,75,7207
175 DATA 9B06,00,5D,DD,74,01,54,DD,70,02,D [2306]
D,71,04,CD,00,9F,C3,0DBE
176 DATA 9B16,C9,97,C5,3E,EC,04,C6,14,10,F [3842]
C,81,5F,16,00,21,96,CAE8
177 DATA 9B26,96,19,7E,C1,C9,65,74,00,12,0 [2818]
0,E2,18,01,C0,69,62,B331
178 DATA 9B36,6C,6F,63,6B,20,64,77,20,30,0 [2884]
0,0C,00,EC,18,01,C0,5065
179 DATA 9B46,64,77,20,32,30,00,0C,00,F6,1 [2714]
8,01,C0,64,77,20,31,0B84
180 DATA 9B56,35,00,0B,00,00,19,01,C0,64,7 [2454]
7,20,30,00,0B,00,0A,6EFD
181 DATA 9B66,19,01,C0,64,62,20,30,00,0D,0 [2719]
0,14,19,01,C0,64,62,7623
182 DATA 9B76,20,26,38,31,00,10,00,1E,19,0 [3099]
1,C0,64,77,20,61,75,DF12
183 DATA 9B86,73,77,61,62,00,1A,00,28,19,0 [2705]
1,C0,69,6E,74,6F,75,1075
184 DATA 9B96,74,20,6C,64,20,68,6C,2C,69,6 [4204]
2,6C,6F,63,6B,00,11,B1EC
185 DATA 9BA6,00,32,19,01,C0,63,61,6C,6C,2 [2451]
0,26,62,63,65,63,00,ACA2
186 DATA 9BB6,13,00,3C,19,01,C0,6C,64,20,6 [2169]
8,6C,2C,69,62,6C,6F,233B
187 DATA 9BC6,63,6B,00,0F,00,46,19,01,C0,6 [3372]
A,70,20,26,62,63,66,EDB2
188 DATA 9BD6,38,00,00,00,1A,00,00,00,00,0 [3546]
0,00,00,00,00,00,00,D01C
189 DATA 9BE6,00,00,00,00,00,00,00,00,00,0 [3936]
0,00,00,00,00,00,00,0000
190 DATA 9BF6,00,00,00,00,00,00,00,A6,13,0 [3438]
1,C0,63,61,6C,6C,20,B3B0
191 DATA 9C06,76,00,10,00,B0,13,01,C0,6C,6 [3714]
4,20,61,2C,28,79,76,2268
192 DATA 9C16,29,00,0C,00,BA,13,01,C0,69,6 [2536]
E,63,20,61,00,0C,00,7C60
193 DATA 9C26,C4,13,01,C0,3E,01,CD,AD,BC,E [3075]
6,07,28,20,3A,46,9E,4770
194 DATA 9C36,3C,FE,05,20,01,97,32,46,9E,1 [3394]
6,00,5F,62,6B,29,29,1B96
195 DATA 9C46,29,19,11,07,9E,19,DD,E5,CD,A [2355]
A,BC,DD,E1,3A,F5,9D,0BD2
196 DATA 9C56,3C,32,F5,9D,FE,09,C0,97,32,F [2111]
5,9D,3A,F2,9D,47,3A,0213
197 DATA 9C66,F3,9D,B8,C8,3A,F2,9D,5F,16,0 [2526]
0,21,26,9D,19,7E,32,3444
198 DATA 9C76,F0,9D,3A,F2,9D,5F,16,00,21,8 [3630]
B,9D,19,7E,32,F1,9D,BA65
```

Listing LETSWORK

```
199 DATA 9C86,3A,F2,9D,3C,FE,64,20,01,97,3 [3819]
2,F2,9D,3A,F0,9D,3D,92B0
200 DATA 9C96,32,F0,9D,CD,C4,9C,3A,F0,9D,3 [3534]
C,3C,32,F0,9D,CD,C4,E5B4
201 DATA 9CA6,9C,3A,F0,9D,3D,32,F0,9D,3A,F [2634]
1,9D,3D,32,F1,9D,CD,3FC4
202 DATA 9CB6,C4,9C,3A,F1,9D,3C,3C,32,F1,9 [3742]
D,CD,C4,9C,C9,ED,4B,4E3C
203 DATA 9CC6,F0,9D,CD,18,9B,B7,C0,3A,86,9 [3597]
7,B9,20,0D,3A,87,97,ACB9
204 DATA 9CD6,B8,20,07,D1,3E,7B,32,89,97,C [2705]
9,DD,71,04,DD,70,02,679F
205 DATA 9CE6,11,80,A2,C5,CD,00,9F,C1,CD,1 [3858]
8,9B,36,05,3A,F3,9D,760C
206 DATA 9CF6,5F,16,00,21,26,9D,19,3A,F0,9 [2920]
D,77,3A,F3,9D,5F,16,0B6F
207 DATA 9D06,00,21,8B,9D,19,3A,F1,9D,77,3 [3055]
A,F3,9D,3C,FE,64,20,8F51
208 DATA 9D16,01,97,32,F4,9D,47,3A,F2,9D,B [3066]
8,C8,78,32,F3,9D,C9,4F86
209 DATA 9D26,6E,6F,6E,69,74,20,6C,64,20,2 [3313]
8,69,78,2B,34,29,2C,1D72
210 DATA 9D36,63,00,12,00,82,14,01,C0,6C,6 [3157]
4,20,28,69,78,2B,32,697B
211 DATA 9D46,29,2C,62,00,12,00,8C,14,01,C [3454]
0,6C,64,20,64,65,2C,6724
212 DATA 9D56,26,61,32,38,30,00,0E,00,96,1 [3055]
4,01,C0,70,75,73,68,5CF5
213 DATA 9D66,20,62,63,00,11,00,A0,14,01,C [3751]
0,63,61,6C,6C,20,26,0443
214 DATA 9D76,39,66,30,30,00,0D,00,AA,14,0 [3225]
1,C0,70,6F,70,20,62,89DA
215 DATA 9D86,63,00,0E,00,B4,14,01,C0,63,6 [2495]
1,6C,6C,20,71,75,00,509B
216 DATA 9D96,10,00,BE,14,01,C0,6C,64,20,2 [3199]
8,68,6C,29,2C,35,00,244F
217 DATA 9DA6,10,00,C8,14,01,C0,6C,64,20,6 [2240]
1,2C,28,7A,32,29,00,F819
218 DATA 9DB6,0D,00,D2,14,01,C0,6C,64,20,6 [3064]
5,2C,61,00,0D,00,DC,3E26
219 DATA 9DC6,14,01,C0,6C,64,20,64,2C,30,0 [2198]
0,0F,00,E6,14,01,C0,DA55
220 DATA 9DD6,6C,64,20,68,6C,2C,78,61,00,1 [3288]
0,00,F0,14,01,C0,61,CB6B
221 DATA 9DE6,64,64,20,68,6C,2C,64,65,00,1 [1609]
0,00,00,00,00,00,00,F92A
222 DATA 9DF6,02,08,01,01,05,FF,02,03,02,0 [2653]
2,02,02,FE,02,0A,05,E509
223 DATA 9E06,01,01,01,01,DE,01,02,08,1E,0 [3716]
0,01,01,01,AA,01,00,0594
224 DATA 9E16,06,1E,00,01,01,01,7B,01,01,0 [3157]
6,1E,00,01,01,01,AA,6960
225 DATA 9E26,01,00,06,1E,00,01,01,01,DE,0 [2426]
1,00,06,1E,00,02,00,C8D5
226 DATA 9E36,00,00,00,01,0A,1A,00,02,00,0 [3021]
0,00,00,14,0C,05,00,2A9A
227 DATA 9E46,00,21,3D,9E,DD,E5,CD,AA,BC,D [2987]
D,E1,C3,1E,98,DD,E5,369D
228 DATA 9E56,21,34,9E,CD,AA,BC,DD,E1,C9,0 [2679]
0,00,14,00,0F,00,00,0EF8
229 DATA 9E66,00,00,81,2A,9C,21,5F,9E,CD,E [3356]
C,BC,21,5F,9E,C3,F8,8DF8
230 DATA 9E76,BC,21,A7,A1,06,06,E5,11,E0,C [4203]
1,CD,99,9E,11,E0,C9,1684
231 DATA 9E86,CD,99,9E,CD,BF,9E,E1,10,ED,1 [4023]
1,14,00,19,7D,FE,F7,F22E
232 DATA 9E96,20,E2,C9,C5,06,0A,3E,01,32,F [1804]
F,9E,4E,C5,3A,FF,9E,A37D
233 DATA 9EA6,A1,0E,00,28,02,0E,F0,79,12,3 [1873]
A,FF,9E,87,32,FF,9E,2540
234 DATA 9EB6,C1,13,20,E8,23,10,DF,C1,C9,C [2550]
5,11,00,C0,21,00,C8,B6C2
235 DATA 9EC6,01,50,00,E5,ED,B0,D1,62,6B,3 [3868]
E,08,84,67,30,08,3E,8888
236 DATA 9ED6,50,85,6F,3E,C0,8C,67,FE,C9,2 [2937]
0,E5,7D,FE,E0,20,E0,3323
237 DATA 9EE6,21,D0,C2,11,D0,CA,01,50,00,E [3210]
5,ED,B0,D1,6B,62,CD,02F5
238 DATA 9EF6,29,BC,7A,FE,C9,20,EF,C1,C9,0 [3242]
0,D5,DD,6E,02,26,00,87F8
239 DATA 9F06,29,29,29,29,29,54,5D,29,29,1 [2616]
9,16,C0,DD,7E,04,87,E086
240 DATA 9F16,87,5F,19,D1,EB,01,04,10,C5,D [3750]
5,06,00,ED,B0,D1,C1,6796
241 DATA 9F26,CD,45,9F,10,F3,C9,DD,6E,04,2 [2388]
6,00,29,29,29,29,29,40E7
242 DATA 9F36,54,5D,29,29,19,16,C0,DD,7E,0 [3839]
6,87,87,5F,19,C9,3E,181B
```

Listing LETSWORK

```
243 DATA 9F46,08,82,57,D0,3E,50,83,5F,3E,C [3440]
0,8A,57,C9,CD,2C,9F,B7A9
244 DATA 9F56,EB,DD,7E,02,B7,01,80,A4,28,0 [2777]
3,01,C0,A4,21,81,9F,2490
245 DATA 9F66,71,23,70,97,32,E5,9F,32,E6,9 [2933]
F,D5,CD,19,BD,CD,19,793B
246 DATA 9F76,BD,CD,19,BD,CD,19,BD,CD,19,B [3010]
D,21,00,00,3A,E6,9F,AD39
247 DATA 9F86,85,6F,06,10,7E,12,23,23,23,2 [3218]
3,CD,45,9F,10,F5,D1,5C68
248 DATA 9F96,13,D5,3A,E5,9F,EE,01,32,E5,9 [3911]
F,21,80,A3,20,03,21,FCED
249 DATA 9FA6,C0,A3,01,04,10,C5,D5,06,00,E [4044]
D,B0,D1,C1,CD,45,9F,BD33
250 DATA 9FB6,10,F3,DD,7E,00,B7,D1,D5,28,1 [2322]
7,13,13,13,13,21,40,396D
251 DATA 9FC6,A2,01,04,10,C5,D5,06,00,ED,B [2362]
0,D1,C1,CD,45,9F,10,FAB7
252 DATA 9FD6,F3,D1,3A,E6,9F,3C,FE,04,32,E [2563]
6,9F,C2,70,9F,C9,00,2F6F
253 DATA 9FE6,00,CD,2C,9F,EB,DD,7E,02,B7,0 [1532]
1,80,A4,28,03,01,C0,A235
254 DATA 9FF6,A4,21,1A,A0,71,23,70,97,32,E [1911]
5,9F,3E,03,32,E6,9F,4293
255 DATA A006,13,13,13,D5,CD,19,BD,CD,19,B [3917]
D,CD,19,BD,CD,19,BD,F6A6
256 DATA A016,CD,19,BD,21,00,00,3A,E6,9F,8 [3679]
5,6F,06,10,7E,12,23,804A
257 DATA A026,23,23,23,CD,45,9F,10,F5,D1,1 [2832]
B,1B,1B,1B,D5,3A,E5,1E9C
258 DATA A036,9F,EE,01,32,E5,9F,21,00,A4,2 [3052]
0,03,21,40,A4,01,04,4E94
259 DATA A046,10,C5,D5,06,00,ED,B0,D1,C1,C [2443]
D,45,9F,10,F3,DD,7E,D539
260 DATA A056,00,B7,D1,D5,28,17,1B,1B,1B,1 [3054]
B,21,40,A2,01,04,10,8F57
261 DATA A066,C5,D5,06,00,ED,B0,D1,C1,CD,4 [2996]
5,9F,10,F3,D1,13,13,AA9A
262 DATA A076,13,3A,E6,9F,3D,32,E6,9F,3C,C [3859]
2,09,A0,C9,CD,2C,9F,6309
263 DATA A086,CD,29,BC,11,A0,00,AF,ED,52,E [3329]
B,DD,7E,02,B7,01,80,EEB0
264 DATA A096,A4,28,03,01,C0,A4,21,FB,A0,7 [2657]
1,23,70,97,32,E5,9F,1359
265 DATA A0A6,3E,3C,32,E6,9F,D5,CD,19,BD,C [2986]
D,19,BD,DD,7E,00,B7,6D89
266 DATA A0B6,CA,CC,A0,21,40,A2,01,04,10,C [3261]
5,D5,06,00,ED,B0,D1,BFC5
267 DATA A0C6,C1,CD,45,9F,10,F3,D1,D5,21,A [2486]
0,00,19,EB,3A,E5,9F,94A5
268 DATA A0D6,EE,01,32,E5,9F,21,00,A3,28,0 [2119]
3,21,40,A3,01,04,10,9E9F
269 DATA A0E6,C5,D5,06,00,ED,B0,D1,C1,CD,4 [2073]
5,9F,10,F3,D1,D5,21,AB24
270 DATA A0F6,40,01,19,EB,21,00,00,3A,E6,9 [2833]
F,06,00,4F,09,01,04,B476
271 DATA A106,00,ED,B0,E1,CD,29,BC,EB,3A,E [2811]
6,9F,D6,04,32,E6,9F,349F
272 DATA A116,D2,AB,A0,C9,CD,2C,9F,EB,DD,7 [3200]
E,02,B7,01,80,A4,28,2404
273 DATA A126,03,01,C0,A4,21,3F,A1,71,23,7 [3629]
0,97,32,E5,9F,32,E6,5E85
274 DATA A136,9F,D5,CD,19,BD,CD,19,BD,21,0 [2989]
0,00,3A,E6,9F,4F,06,B590
275 DATA A146,00,09,01,04,00,ED,B0,D1,D5,C [3486]
D,45,9F,3A,E5,9F,EE,7E04
276 DATA A156,01,32,E5,9F,21,00,A3,28,03,2 [3048]
1,40,A3,01,04,10,C5,3D14
277 DATA A166,D5,06,00,ED,B0,D1,C1,CD,45,9 [3169]
F,10,F3,D1,D5,DD,7E,570A
278 DATA A176,00,B7,28,1B,CD,45,9F,21,A0,0 [3188]
0,19,EB,21,40,A2,01,4FF2
279 DATA A186,04,10,D5,C5,06,00,ED,B0,C1,D [4049]
1,CD,45,9F,10,F3,D1,E6AE
280 DATA A196,CD,45,9F,3A,E6,9F,C6,04,32,E [3306]
6,9F,FE,40,C2,37,A1,7924
281 DATA A1A6,C9,87,DF,DF,FD,00,87,FB,F7,E [3108]
F,38,BF,1F,07,FC,00,00F9
282 DATA A1B6,FF,FB,77,EE,38,87,DF,DF,FD,0 [2595]
0,87,FB,F7,EF,1C,BF,9810
283 DATA A1C6,1F,07,FC,00,FF,FB,77,E7,1C,8 [3484]
7,03,07,1C,00,87,3B,F310
284 DATA A1D6,77,EC,0E,87,03,07,E0,00,B7,3 [3849]
B,F7,EF,0E,87,0F,07,B32D
285 DATA A1E6,FC,00,87,3B,77,EC,03,87,0F,0 [3335]
7,FC,00,B7,3B,F7,EF,C4F8
286 DATA A1F6,03,00,00,00,00,00,00,00,00,0 [3871]
0,0C,0F,0C,0F,0C,0F,813A
287 DATA A206,0C,0F,0C,0F,0C,0F,00,00,00,0 [2071]
```

Listing LETSWORK

```
0,0F,03,0F,03,0F,03,EDBD
288 DATA A216,0F,03,0F,03,0F,03,00,00,00,0 [3665]
0,03,0F,03,0F,03,0F,E4BB
289 DATA A226,03,0F,03,0F,03,0F,00,00,00,0 [2036]
0,0F,0C,0F,0C,0F,0C,F578
290 DATA A236,0F,0C,0F,0C,0F,0C,00,00,00,0 [2628]
0,00,00,00,00,F5,F5,E91A
291 DATA A246,F5,F5,FA,FA,FA,FA,3D,F5,F5,F [2621]
4,9F,FF,FF,F8,3F,FF,C85A
292 DATA A256,FF,F8,9F,FF,FF,F8,3F,FF,FF,F [3538]
8,9F,FF,FF,F8,3F,FF,5050
293 DATA A266,FF,F8,9F,FF,FF,F8,3F,FF,FF,F [2688]
8,8F,0F,0F,3C,0F,0F,5950
294 DATA A276,0F,1E,0F,0F,0F,0F,00,00,00,0 [3630]
0,1E,F0,0F,0F,78,F0,5885
295 DATA A286,87,0F,F0,E1,0F,0F,F0,87,0F,1 [3569]
F,F0,E1,0F,7F,F0,0F,636A
296 DATA A296,1F,FF,FF,CF,0F,7F,FF,EF,1F,F [4106]
F,FF,8F,0F,FF,FE,87,E08C
297 DATA A2A6,0F,3F,FC,E1,0F,1F,F0,F0,87,0 [3120]
F,F0,E1,0F,0F,F0,0F,95E3
298 DATA A2B6,0F,0F,3C,87,0F,0F,1E,C3,0F,0 [4573]
F,00,30,C0,00,00,F0,0ABB
299 DATA A2C6,F0,C0,00,6F,CE,00,00,0B,0A,0 [3741]
0,00,04,04,00,00,03,8E2B
300 DATA A2D6,08,00,31,FC,FC,C8,60,FC,FC,6 [2725]
0,66,F3,F3,66,66,F3,7FC6
301 DATA A2E6,F3,66,60,FC,FC,60,06,FC,FC,0 [2474]
6,00,5A,5A,00,00,4A,28DF
302 DATA A2F6,25,00,00,A4,52,00,03,0E,07,0 [3139]
C,00,30,C0,00,00,F0,5D6A
303 DATA A306,F0,C0,00,6F,CE,00,00,0B,0A,0 [2502]
0,00,04,04,00,00,03,8E2B
304 DATA A316,08,00,31,FC,FC,C8,60,FC,FC,6 [3240]
0,66,F3,F3,66,66,F3,7FC6
305 DATA A326,F3,66,60,FC,FC,60,06,FC,FC,0 [1572]
6,00,5A,5A,00,00,A4,2831
306 DATA A336,25,00,03,0E,52,00,00,00,07,0 [3408]
C,00,30,C0,00,00,F0,9560
307 DATA A346,F0,C0,00,6F,CE,00,00,0B,0A,0 [2343]
0,00,04,04,00,00,03,8E2B
308 DATA A356,08,00,31,FC,FC,C8,60,FC,FC,6 [3129]
0,66,F3,F3,66,66,F3,7FC6
309 DATA A366,F3,66,60,FC,FC,60,06,FC,FC,0 [3678]
6,00,5A,5A,00,00,A4,2831
310 DATA A376,25,00,00,4A,07,0C,03,0E,00,0 [2698]
0,00,10,E0,00,00,70,2666
311 DATA A386,F0,E0,00,61,6F,00,00,43,0F,0 [2672]
0,00,43,0F,00,00,01,288C
312 DATA A396,0C,00,00,10,EC,00,00,30,FC,0 [3373]
0,00,33,F3,00,00,33,2A9B
313 DATA A3A6,F3,00,00,30,FC,00,00,30,3C,0 [2912]
0,00,12,5A,00,00,21,4DAC
314 DATA A3B6,A5,00,00,10,4A,00,00,01,0F,0 [3830]
0,00,10,E0,00,00,70,D0A1
315 DATA A3C6,F0,E0,00,61,6F,00,00,43,0F,0 [2619]
0,00,43,0E,00,00,01,2884
316 DATA A3D6,0C,00,00,10,EC,00,00,30,FC,0 [3265]
0,00,77,F3,88,00,F3,2C3B
317 DATA A3E6,F3,C0,11,B8,FC,66,03,30,FC,0 [3236]
3,00,21,A5,00,00,52,17EC
318 DATA A3F6,21,80,00,A4,10,48,00,0F,09,0 [1640]
F,00,70,80,00,70,F0,6B6B
319 DATA A406,E0,00,00,6F,68,00,00,0F,2C,0 [3214]
0,00,07,2C,00,00,03,A866
320 DATA A416,08,00,00,73,80,00,00,F3,C0,0 [2940]
0,00,FC,CC,00,00,FC,AA5B
321 DATA A426,CC,00,00,F3,C0,00,00,C3,C0,0 [3031]
0,00,5A,48,00,00,A5,942A
322 DATA A436,84,00,00,52,08,00,00,0F,08,0 [2930]
0,00,70,80,00,70,F0,6857
323 DATA A446,E0,00,00,6F,68,00,00,0F,2C,0 [3803]
0,00,07,2C,00,00,03,A866
324 DATA A456,08,00,00,73,80,00,00,F3,C0,0 [3460]
0,11,FC,EE,00,30,FC,A90B
325 DATA A466,FC,00,66,F3,D1,88,0C,F3,C0,0 [3351]
C,00,5A,48,00,10,48,D7F1
326 DATA A476,A4,00,21,80,52,00,0F,09,0F,0 [2651]
0,00,00,00,00,00,0C,A0D0
327 DATA A486,00,00,00,00,00,66,00,00,00,0 [3228]
0,60,00,0C,01,00,00,9465
328 DATA A496,00,CC,00,08,00,00,00,00,00,0 [2130]
0,00,00,C0,00,03,00,8635
329 DATA A4A6,00,00,00,00,00,00,00,66,00,0 [1940]
0,00,00,00,00,00,00,6600
330 DATA A4B6,00,80,06,00,00,00,00,00,C8,0 [3298]
0,F0,F0,F0,F0,78,F0,B160
331 DATA A4C6,F0,F1,BC,F0,F0,F2,5E,F0,F0,F [2603]
```

Listing LETSWORK

```
5,AF,F0,F0,FA,5F,78,9F56
332 DATA A4D6,F1,F5,AF,BC,F2,FA,5F,5E,F5,F [2969]
5,AF,AF,7A,FA,5F,4F,2046
333 DATA A4E6,3D,F5,AF,8F,1E,FA,5F,0F,0F,F [1966]
5,AF,0F,0F,7A,4F,0F,57EC
334 DATA A4F6,0F,3D,8F,0F,0F,1E,0F,0F,0F,0 [2466]
F,00,00,00,00,00,00,C559
335 DATA *ENDE*                           [502]
336 adr=&9696:zeile=104:MEMORY adr-1      [1841]
337 READ d$:IF d$="*ENDE*"THEN 348        [1116]
338 pr=0                                  [117]
339 FOR i=1 TO 16                         [317]
340 READ a$:a=VAL("&"+a$)                 [883]
341 POKE adr,a:adr=adr+1                  [717]
342 pr=pr*2:IF pr>65535 THEN pr=pr-65535  [2485]
343 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+655 [1890]
36
344 NEXT i                                [375]
345 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THE [2610]
N pr2=pr2+65536
346 IF pr<>pr2 THEN PRINT"Pruefsummenfehle [3615]
r in Zeile";zeile:STOP
347 zeile=zeile+1:GOTO 337                [987]
348 SAVE"LETSWORK.BIN",B,&9696,&E6A       [2430]
349 PRINT d$:END                          [346]
```

```
100 '*********************************** [1460]
101 '* CLBINLAD - DATA-Lader von 'CPC' *  [2749]
102 '*********************************** [1460]
103 '                                     [117]
104 DATA 6000,01,DE,15,02,DE,04,00,02,DE,0 [1730]
D,00,02,02,DE,04,00,0D33
105 DATA 6010,02,00,02,00,02,01,02,00,00,D [2726]
E,04,02,00,02,02,00,672D
106 DATA 6020,01,00,00,02,DE,07,00,01,01,0 [3524]
0,00,02,00,02,02,00,4DAA
107 DATA 6030,01,00,01,02,00,00,02,00,02,0 [3297]
0,00,01,00,00,00,02,8512
108 DATA 6040,00,02,02,02,00,02,DE,0F,00,0 [1915]
2,02,02,00,02,01,00,5BEB
109 DATA 6050,02,00,00,DE,04,02,00,00,00,0 [3111]
2,00,DE,04,02,00,02,C546
110 DATA 6060,00,02,02,02,DE,06,00,02,00,0 [2686]
0,02,80,02,02,00,01,025F
111 DATA 6070,DE,07,00,02,02,DE,04,00,02,8 [2456]
0,80,02,02,00,02,02,B15B
112 DATA 6080,02,00,00,02,00,00,00,01,00,0 [2282]
2,00,00,80,80,80,02,2683
113 DATA 6090,02,DE,06,00,02,DE,05,00,02,0 [3232]
0,DE,04,80,DE,15,02,3FE5
114 DATA 60A0,25,03,FF,FF,00,01,DE,06,00,D [4297]
E,0D,02,DE,06,00,02,DEE9
115 DATA 60B0,02,80,80,DE,04,00,01,00,02,0 [2981]
0,00,02,DE,05,00,02,C5FA
116 DATA 60C0,02,80,80,DE,08,00,01,00,02,D [4376]
E,04,00,02,02,80,80,95A4
117 DATA 60D0,DE,04,00,02,02,02,DE,04,00,0 [2372]
2,DE,04,00,02,80,80,9AE7
118 DATA 60E0,DE,04,00,02,00,00,00,02,02,0 [2869]
0,02,02,00,DE,04,02,207C
119 DATA 60F0,DE,05,00,02,00,00,01,DE,05,0 [3391]
0,02,00,02,00,00,02,BEBC
120 DATA 6100,02,00,00,00,02,00,00,01,00,0 [3090]
0,01,02,01,00,02,00,110D
121 DATA 6110,02,DE,04,00,02,00,00,00,DE,0 [3072]
4,02,00,00,02,00,00,7E7E
122 DATA 6120,02,00,02,02,02,00,01,DE,04,0 [3630]
0,02,00,00,02,00,00,AE49
123 DATA 6130,02,00,00,02,00,00,00,02,00,0 [2535]
1,02,00,02,01,02,00,2211
124 DATA 6140,00,02,00,02,02,02,01,02,00,0 [3455]
0,00,02,00,00,02,DE,B8FA
125 DATA 6150,09,00,01,00,00,00,02,00,00,D [2430]
E,12,02,03,0D,FF,FF,90C9
126 DATA 6160,00,01,DE,15,02,DE,04,00,02,0 [2980]
0,00,00,02,00,80,80,B089
127 DATA 6170,00,00,02,00,00,00,02,02,00,0 [3447]
2,02,00,02,00,01,00,46D2
128 DATA 6180,00,00,80,80,02,DE,05,00,02,0 [2810]
2,00,00,02,00,02,DE,6351
129 DATA 6190,04,00,02,80,80,02,00,01,02,0 [2690]
1,00,02,02,00,00,02,487C
130 DATA 61A0,00,02,DE,07,00,02,DE,05,00,0 [3014]
2,02,00,00,02,00,02,81D0
131 DATA 61B0,02,00,DE,04,02,DE,07,00,02,0 [3327]
2,01,02,DE,0A,00,DE,E19F
```

Listing LETSWORK

```
132 DATA 61C0,05,02,00,02,02,00,02,02,02,0 [4208]
0,02,DE,04,00,02,DE,3A58
133 DATA 61D0,07,00,02,02,00,00,02,00,00,0 [3096]
1,00,00,01,00,02,00,E44F
134 DATA 61E0,00,00,01,02,00,00,02,02,01,0 [2965]
0,02,00,01,00,02,02,06CE
135 DATA 61F0,80,DE,04,00,02,00,00,01,00,0 [2886]
2,02,00,00,02,00,00,11BF
136 DATA 6200,00,01,80,80,00,02,DE,07,00,D [3109]
E,15,02,03,15,FF,FF,C754
137 DATA 6210,00,01,DE,15,02,DE,06,00,DE,0 [3487]
8,80,DE,04,00,02,02,C4DF
138 DATA 6220,DE,12,00,02,02,DE,05,00,DE,0 [3359]
5,01,DE,08,00,02,02,A1AE
139 DATA 6230,DE,05,00,01,00,01,00,01,DE,0 [1911]
8,00,02,02,DE,05,00,3B2C
140 DATA 6240,DE,05,01,DE,08,00,02,02,DE,0 [2759]
5,00,01,00,00,00,01,A832
141 DATA 6250,DE,08,00,02,02,DE,05,00,DE,0 [2943]
5,01,DE,08,00,02,02,21A8
142 DATA 6260,DE,12,00,02,02,DE,12,00,02,0 [2206]
2,DE,04,00,02,DE,05,F7D9
143 DATA 6270,00,02,DE,07,00,DE,15,02,15,1 [2307]
5,FF,FF,00,01,DE,15,7E65
144 DATA 6280,02,DE,12,00,02,02,00,02,00,0 [3152]
2,00,DE,09,02,00,02,D716
145 DATA 6290,02,00,02,02,00,01,00,02,00,0 [3018]
2,DE,08,80,01,00,02,79C7
146 DATA 62A0,00,02,02,00,02,00,02,00,DE,0 [3191]
4,02,00,DE,04,02,00,BCA4
147 DATA 62B0,00,02,00,02,02,DE,10,00,02,0 [3309]
0,02,02,00,01,00,00,E967
148 DATA 62C0,01,DE,04,00,02,00,01,02,00,0 [2892]
0,00,02,00,02,02,00,901B
149 DATA 62D0,02,00,02,DE,05,00,02,01,00,0 [1095]
2,80,00,00,02,00,02,9D86
150 DATA 62E0,02,00,02,01,02,01,DE,05,02,0 [3289]
0,01,02,80,00,00,02,F802
151 DATA 62F0,00,02,02,00,02,00,02,DE,06,0 [2635]
0,01,00,02,02,02,00,093C
152 DATA 6300,00,00,02,02,DE,12,00,DE,15,0 [4095]
2,17,03,FF,FF,00,01,0AD3
153 DATA 6310,DE,15,02,DE,10,00,02,00,02,0 [3142]
2,DE,04,00,01,02,DE,7EB9
154 DATA 6320,04,00,01,02,00,01,00,00,02,0 [2722]
0,02,02,00,00,02,00,0566
155 DATA 6330,00,02,00,01,02,00,00,02,DE,0 [3122]
6,00,02,02,00,02,ECB2
156 DATA 6340,00,01,02,00,00,02,00,01,02,0 [3394]
0,DE,04,80,00,02,02,1786
157 DATA 6350,00,01,02,00,00,02,00,01,02,0 [2110]
0,00,02,00,DE,04,80,0BD0
158 DATA 6360,00,02,02,00,00,02,00,01,02,0 [2893]
0,00,02,00,01,02,00,C820
159 DATA 6370,DE,04,80,00,02,02,00,01,02,0 [1767]
0,00,02,00,01,02,00,185E
160 DATA 6380,00,02,DE,06,00,02,02,00,00,0 [2803]
2,00,01,02,00,00,02,2C99
161 DATA 6390,00,01,02,DE,06,00,02,02,00,0 [3034]
1,02,DE,04,00,01,02,DBCD
162 DATA 63A0,DE,09,00,02,02,DE,12,00,DE,1 [2525]
5,02,25,03,25,15,00,4498
163 DATA 63B0,01,DE,15,02,DE,05,00,02,00,0 [2289]
2,02,00,00,02,00,00,66FB
164 DATA 63C0,00,80,02,80,02,02,00,02,00,0 [3374]
2,00,02,00,01,00,00,5A8C
165 DATA 63D0,00,02,00,02,02,00,02,81,02,0 [2660]
2,00,02,01,02,00,02,34A2
166 DATA 63E0,00,02,02,00,00,02,00,02,00,0 [2664]
0,02,80,02,02,00,00,C258
167 DATA 63F0,00,01,00,01,DE,04,00,02,02,D [2614]
E,04,00,01,00,02,02,8408
168 DATA 6400,00,02,00,DE,04,02,DE,07,00,0 [3592]
2,00,00,00,02,02,00,F380
169 DATA 6410,02,00,02,DE,05,00,02,DE,04,0 [3048]
0,02,00,00,02,02,02,5042
170 DATA 6420,00,02,00,02,00,00,02,00,00,0 [2205]
2,00,01,01,00,00,02,A49A
171 DATA 6430,00,00,02,02,00,02,00,02,00,0 [2658]
1,00,00,81,80,DE,08,6DFC
172 DATA 6440,00,02,02,00,02,00,02,00,DE,0 [3425]
8,02,00,00,02,00,80,B9C8
173 DATA 6450,02,02,00,02,00,02,00,00,01,D [3221]
E,05,00,80,00,80,02,9AA3
174 DATA 6460,80,80,DE,15,02,07,15,FF,FF,0 [3156]
0,01,02,02,02,00,DE,363C
175 DATA 6470,05,02,DE,0B,00,02,00,00,02,0 [3699]
0,00,01,00,00,02,00,790D
```

Listing LETSWORK

```
176 DATA 6480,00,02,02,02,DE,05,00,02,00,0 [3312]
1,00,00,00,02,00,00,064E
177 DATA 6490,00,02,00,02,00,00,02,DE,04,0 [2675]
0,02,02,02,00,00,02,7872
178 DATA 64A0,DE,05,00,02,DE,04,00,DE,04,0 [2270]
2,00,00,02,00,01,DE,5C24
179 DATA 64B0,05,00,01,02,00,01,01,01,00,0 [2641]
0,00,02,00,00,00,02,8720
180 DATA 64C0,DE,07,00,02,00,02,00,00,00,0 [3206]
1,00,02,DE,04,00,02,EEEC
181 DATA 64D0,DE,09,00,02,02,00,00,00,02,D [3772]
E,04,00,02,80,DE,0A,45CB
182 DATA 64E0,00,02,02,00,DE,06,02,80,DE,0 [3172]
5,00,02,02,02,00,01,C27F
183 DATA 64F0,01,00,02,02,02,DE,05,80,DE,0 [2200]
4,00,01,02,DE,07,00,6F75
184 DATA 6500,02,02,80,80,80,DE,04,00,01,D [1761]
E,05,00,01,00,00,01,C7B7
185 DATA 6510,00,DE,15,02,03,03,FF,FF,00,0 [2652]
1,DE,15,02,DE,08,00,0C9C
186 DATA 6520,DE,0A,80,02,02,DE,12,00,02,0 [2850]
2,DE,12,00,02,02,00,F712
187 DATA 6530,00,00,DE,05,01,DE,0A,00,02,0 [3733]
2,00,00,00,01,00,01,F59D
188 DATA 6540,00,01,DE,0A,00,02,02,00,00,0 [3387]
0,01,01,00,00,01,DE,2CF7
189 DATA 6550,0A,00,02,02,00,00,00,01,00,0 [3586]
1,00,01,DE,0A,00,02,678F
190 DATA 6560,02,00,00,00,DE,05,01,DE,0A,0 [3447]
0,02,02,DE,12,00,02,3BDD
191 DATA 6570,02,DE,12,00,DE,15,02,03,15,1 [3121]
9,07,00,01,DE,15,02,6C4A
192 DATA 6580,DE,11,00,80,02,02,80,81,DE,0 [3188]
D,00,01,00,00,02,02,B534
193 DATA 6590,DE,06,00,02,02,01,01,02,02,D [2519]
E,06,00,02,02,DE,05,838F
194 DATA 65A0,00,02,00,00,00,01,00,00,02,D [3747]
E,05,00,02,02,DE,05,B381
195 DATA 65B0,00,02,DE,06,00,02,DE,05,00,0 [4229]
2,02,DE,04,00,02,00,9C1E
196 DATA 65C0,00,00,02,02,00,00,00,02,DE,0 [2769]
4,00,02,02,01,00,00,0C34
197 DATA 65D0,02,DE,04,00,80,02,02,00,00,0 [3790]
0,02,00,01,00,02,02,0C7C
198 DATA 65E0,00,00,02,DE,05,00,02,02,DE,0 [2994]
5,00,02,00,00,02,02,E06B
199 DATA 65F0,00,02,00,02,00,00,02,02,02,8 [3249]
0,DE,06,00,02,00,02,9CAA
200 DATA 6600,02,02,80,DE,06,00,02,02,DE,0 [3770]
6,00,80,DE,16,02,0D,363D
201 DATA 6610,03,FF,FF,00,01,DE,15,02,DE,1 [3178]
2,00,02,02,DE,05,00,90E0
202 DATA 6620,01,00,01,01,DE,09,00,02,02,0 [3418]
0,00,02,00,00,02,02,6720
203 DATA 6630,00,02,02,02,01,02,00,02,02,0 [2758]
0,00,02,02,00,00,02,E332
204 DATA 6640,00,00,02,80,00,00,02,80,00,0 [2739]
0,00,02,80,00,00,02,C02A
205 DATA 6650,02,00,01,02,00,00,02,02,00,0 [2891]
0,02,DE,04,00,02,02,0B87
206 DATA 6660,00,00,02,02,00,00,02,80,00,0 [3069]
2,80,00,00,02,DE,04,F530
207 DATA 6670,00,80,02,00,00,02,02,00,00,0 [2441]
2,02,00,02,02,00,00,4CF8
208 DATA 6680,02,DE,04,00,02,02,00,00,02,0 [2719]
2,DE,0F,00,01,00,00,0282
209 DATA 6690,02,02,DE,12,00,02,02,DE,12,0 [1975]
0,DE,15,02,1F,03,FF,A11E
210 DATA 66A0,FF,00,01,DE,15,02,DE,12,00,0 [2757]
2,02,DE,0F,00,01,00,4329
211 DATA 66B0,00,02,02,00,02,00,00,00,02,0 [1695]
0,02,02,02,00,02,02,D176
212 DATA 66C0,02,00,02,00,02,00,02,00,02,0 [2533]
0,00,00,02,00,02,00,5515
213 DATA 66D0,02,00,02,00,02,00,02,80,02,0 [3126]
0,02,00,02,00,00,00,D551
214 DATA 66E0,02,01,02,00,02,80,02,00,02,0 [2559]
0,02,02,00,00,02,00,1567
215 DATA 66F0,02,00,00,00,02,00,02,00,02,0 [3204]
0,02,02,00,01,02,02,1563
216 DATA 6700,00,00,02,00,02,80,02,80,02,0 [1649]
0,02,00,02,00,02,80,D5D6
```

Listing LETSWORK

```
217 DATA 6710,02,00,02,80,02,00,02,00,00,0 [3074]
2,80,02,00,00,02,02,44AF
218 DATA 6720,02,01,02,00,02,00,02,00,02,0 [2665]
0,02,DE,04,00,01,00,1883
219 DATA 6730,01,00,00,00,01,DE,07,00,02,0 [2922]
2,00,01,DE,0F,00,80,F9DF
220 DATA 6740,DE,15,02,09,0B,FF,FF,00,01,D [4651]
E,1C,02,80,DE,04,00,F9B8
221 DATA 6750,02,02,02,00,80,00,02,02,02,0 [2605]
0,01,80,02,02,02,80,CFB9
222 DATA 6760,DE,04,00,80,02,02,01,01,80,0 [2599]
2,02,02,00,01,00,DE,5B5C
223 DATA 6770,05,02,DE,04,00,02,02,00,00,0 [2728]
0,02,02,02,DE,09,00,8F03
224 DATA 6780,DE,04,02,00,01,00,00,02,02,0 [3704]
2,DE,05,00,01,DE,06,51C0
225 DATA 6790,00,01,00,01,00,00,02,02,00,0 [2357]
0,80,80,DE,04,02,00,48E4
226 DATA 67A0,00,DE,04,02,00,DE,05,02,00,0 [2651]
0,DE,06,02,00,00,02,4B86
227 DATA 67B0,DE,06,00,02,02,02,00,00,00,D [4061]
E,04,02,DE,0A,00,02,8994
228 DATA 67C0,02,02,DE,09,00,01,02,02,DE,0 [2932]
4,00,80,02,02,02,00,B406
229 DATA 67D0,01,DE,07,00,80,DE,1D,02,15,0 [3365]
5,FF,FF,00,01,DE,15,BA4D
230 DATA 67E0,02,DE,10,00,80,80,02,02,00,0 [2369]
0,02,01,02,00,00,01,8673
231 DATA 67F0,01,DE,07,00,80,80,02,02,00,0 [2580]
0,02,00,00,02,00,00,E679
232 DATA 6800,01,DE,06,00,02,80,80,02,02,0 [3314]
0,00,02,00,00,02,00,D310
233 DATA 6810,01,01,DE,06,00,02,02,80,02,0 [2797]
2,00,00,02,00,00,02,ED89
234 DATA 6820,00,01,DE,06,00,01,DE,05,02,0 [3236]
0,00,00,01,02,DE,0C,5DAA
235 DATA 6830,00,80,02,02,00,00,02,80,02,D [2918]
E,07,00,02,80,DE,06,D1EA
236 DATA 6840,02,00,00,DE,05,02,00,00,00,0 [2030]
2,02,02,00,01,80,80,C168
237 DATA 6850,02,02,02,00,01,DE,0C,00,02,0 [2944]
1,DE,10,02,DE,04,00,B0E2
238 DATA 6860,DE,17,02,19,0F,FF,FF,00,01,D [3278]
E,15,02,DE,11,00,80,59DD
239 DATA 6870,02,02,DE,05,00,01,00,02,00,0 [1883]
0,02,02,02,00,00,02,1668
240 DATA 6880,02,00,02,02,00,00,02,02,00,0 [3468]
0,02,00,00,00,02,80,66C5
241 DATA 6890,02,00,00,00,02,00,02,02,00,0 [2949]
0,80,02,00,02,02,00,062D
242 DATA 68A0,00,00,02,DE,05,00,02,00,02,0 [2664]
2,00,00,00,02,DE,04,8C3D
243 DATA 68B0,00,02,00,00,00,01,00,00,00,0 [2520]
2,00,02,02,02,DE,07,8503
244 DATA 68C0,00,02,00,00,00,02,02,02,DE,04,0 [2446]
0,02,02,02,01,00,00,4074
245 DATA 68D0,00,02,00,01,02,01,DE,08,00,0 [2583]
2,02,00,01,00,02,80,304D
246 DATA 68E0,01,DE,05,00,02,00,00,80,02,0 [2920]
0,00,02,02,00,00,DE,31D9
247 DATA 68F0,04,02,DE,05,00,02,00,00,02,0 [3220]
2,02,01,02,02,DE,04,1869
248 DATA 6900,00,80,00,00,00,02,80,01,DE,0 [3279]
6,00,80,DE,15,02,03,6902
249 DATA 6910,03,FF,FF,00,01,DE,15,02,00,0 [3120]
0,DE,05,02,00,00,00,E3A2
250 DATA 6920,DE,05,02,00,00,00,02,02,00,0 [3404]
0,02,DE,08,00,01,DE,0B52
251 DATA 6930,06,00,02,02,00,00,01,01,DE,0 [2902]
5,02,01,DE,06,02,00,0BFF
252 DATA 6940,00,02,02,00,00,01,00,00,00,0 [2183]
1,DE,04,00,01,00,00,DFC4
253 DATA 6950,00,01,00,00,02,02,00,00,00,D [1607]
E,0B,02,00,01,00,00,6EC4
254 DATA 6960,02,02,DE,0F,00,01,00,00,DE,0 [2113]
4,02,DE,0A,00,02,01,D7EF
255 DATA 6970,01,01,00,00,02,02,02,80,80,8 [2673]
0,02,DE,0D,00,02,02,31CE
256 DATA 6980,DE,04,80,02,DE,0D,00,02,02,0 [3690]
2,80,80,80,02,DE,0D,FA41
257 DATA 6990,00,DE,15,02,15,13,FF,FF,00,0 [3275]
1,DE,16,02,DE,10,00,FC9C
```

Listing LETSWORK

```
258 DATA 69A0,80,02,02,80,02,DE,05,00,01,D [3637]
E,0A,00,02,02,00,80,9493
259 DATA 69B0,02,00,00,00,01,00,01,DE,09,0 [1978]
0,02,02,00,00,80,02,D1E3
260 DATA 69C0,00,00,01,01,01,DE,06,00,80,0 [1853]
0,00,02,02,80,00,00,0E33
261 DATA 69D0,80,02,00,00,01,DE,05,00,02,0 [3702]
2,80,DE,05,02,00,00,E603
262 DATA 69E0,00,80,02,DE,06,00,01,02,DE,0 [2324]
4,00,02,02,00,02,00,FE19
263 DATA 69F0,00,00,80,02,00,00,00,01,00,8 [2826]
0,02,00,01,01,00,02,015E
264 DATA 6A00,02,00,00,02,00,00,00,80,02,0 [2519]
0,00,DE,04,02,DE,04,AD71
265 DATA 6A10,00,02,02,DE,0A,00,02,DE,04,0 [2800]
0,01,00,00,02,02,DE,A8FF
266 DATA 6A20,0A,00,02,DE,07,00,DE,15,02,0 [2445]
3,15,19,15,00,01,DE,338D
267 DATA 6A30,15,02,80,81,DE,0B,00,02,02,0 [3449]
0,00,00,DE,05,02,01,C9F5
268 DATA 6A40,00,DE,07,02,00,01,00,80,00,0 [3186]
0,02,02,00,02,80,DE,C581
269 DATA 6A50,05,00,01,00,00,02,02,00,02,0 [2956]
2,00,01,02,02,00,00,AD8A
270 DATA 6A60,DE,04,02,00,02,02,02,00,02,0 [3504]
2,00,00,00,02,00,02,5DE4
271 DATA 6A70,02,01,00,DE,04,02,00,80,DE,0 [2685]
4,00,02,02,02,01,02,6634
272 DATA 6A80,80,02,02,00,00,02,80,DE,05,0 [3112]
2,DE,04,00,02,02,00,0FCD
273 DATA 6A90,DE,04,02,00,01,DE,04,00,02,0 [3719]
2,02,00,02,02,01,02,39B5
274 DATA 6AA0,80,00,00,02,02,02,00,02,02,0 [3310]
0,00,00,02,02,02,00,3B5C
275 DATA 6AB0,02,02,00,02,80,00,00,00,02,0 [2043]
2,80,02,02,00,00,02,B1B7
276 DATA 6AC0,02,00,01,DE,04,00,80,02,01,0 [2214]
1,00,02,02,00,02,02,E2FB
277 DATA 6AD0,DE,05,00,02,DE,04,00,02,02,8 [4142]
0,00,00,DE,15,02,03,A5CB
278 DATA 6AE0,15,FF,FF,00,01,DE,15,02,DE,0 [2310]
A,00,80,01,DE,06,00,9ED5
279 DATA 6AF0,02,02,DE,0A,00,80,01,DE,05,0 [3193]
0,02,02,02,DE,05,00,3D9A
280 DATA 6B00,DE,0F,02,DE,04,00,01,DE,0D,8 [2622]
0,02,02,DE,05,00,02,BC67
281 DATA 6B10,01,02,02,02,01,02,02,02,01,0 [1852]
2,02,00,02,02,00,01,6659
282 DATA 6B20,00,01,DE,08,00,02,DE,05,00,0 [2420]
2,02,01,00,01,00,01,B1CF
283 DATA 6B30,DE,07,00,02,DE,05,00,02,02,0 [2550]
0,01,00,01,DE,08,00,0428
284 DATA 6B40,02,00,01,00,00,00,02,02,00,0 [1959]
0,01,00,01,DE,07,00,255F
285 DATA 6B50,DE,05,02,00,02,02,80,DE,11,0 [2859]
0,DE,15,02,09,03,25,D468
286 DATA 6B60,03,00,01,DE,15,02,80,DE,11,0 [3451]
0,02,02,00,00,00,DE,3633
287 DATA 6B70,0E,02,01,02,02,00,00,00,02,D [2796]
E,0E,00,02,02,00,00,A75F
288 DATA 6B80,00,02,00,01,DE,09,02,01,00,0 [3947]
0,02,02,00,00,00,02,4164
289 DATA 6B90,DE,04,00,81,80,02,80,80,00,0 [3700]
0,00,01,00,02,02,00,987F
290 DATA 6BA0,00,01,DE,0D,02,00,00,02,02,D [3280]
E,07,00,81,00,00,00,7073
291 DATA 6BB0,81,80,02,80,00,01,00,02,02,D [2986]
E,04,00,02,02,00,00,F070
292 DATA 6BC0,00,DE,05,02,DE,04,00,02,02,0 [2813]
1,DE,04,00,DE,05,02,FB81
293 DATA 6BD0,00,00,00,02,DE,04,00,02,02,8 [3035]
0,DE,0A,00,01,00,02,F860
294 DATA 6BE0,DE,04,00,DE,15,02,07,13,25,1 [3215]
5,00,01,DE,15,02,DE,4CCD
295 DATA 6BF0,08,00,02,00,00,00,80,DE,04,0 [2778]
0,80,02,02,DE,08,00,8F5D
296 DATA 6C00,02,00,02,80,01,DE,05,00,02,0 [2672]
2,00,01,01,80,02,00,3996
297 DATA 6C10,00,00,02,00,02,00,02,00,00,0 [3068]
1,00,00,02,02,00,00,5458
```

Listing LETSWORK

```
298 DATA 6C20,02,80,02,02,02,00,02,00,02,0 [3548]
0,00,01,DE,04,00,02,73D3
299 DATA 6C30,02,00,01,01,80,02,00,02,00,0 [2970]
2,00,02,00,00,DE,07,3B1E
300 DATA 6C40,02,00,00,02,80,02,00,02,00,0 [2675]
2,00,02,02,02,DE,04,2B05
301 DATA 6C50,00,80,02,02,00,01,01,80,02,0 [2458]
0,02,DE,05,00,01,00,EAAA
302 DATA 6C60,00,01,00,00,02,02,00,00,02,8 [2716]
0,02,00,DE,05,02,00,7FA0
303 DATA 6C70,00,00,02,02,80,00,DE,07,02,0 [2505]
0,02,00,00,00,DE,07,DBFE
304 DATA 6C80,02,00,02,02,00,80,DE,06,00,0 [2428]
1,DE,09,00,DE,15,02,C242
305 DATA 6C90,03,15,0D,0B,00,01,DE,15,02,0 [3522]
0,02,80,DE,0A,00,01,729D
306 DATA 6CA0,00,00,00,80,02,02,00,02,00,0 [3672]
0,DE,0C,02,00,00,02,011A
307 DATA 6CB0,02,00,02,00,02,80,DE,0A,00,8 [3440]
0,02,00,02,02,01,02,C65A
308 DATA 6CC0,00,02,00,01,DE,08,02,00,00,0 [2153]
2,00,02,02,00,02,00,44B2
309 DATA 6CD0,02,DE,05,00,01,00,81,80,80,0 [3879]
2,00,02,00,02,02,00,EA9B
310 DATA 6CE0,02,00,00,02,00,00,01,00,00,0 [2930]
0,01,80,80,02,00,02,2E2B
311 DATA 6CF0,00,02,02,00,00,02,00,DE,07,0 [2943]
2,00,00,00,01,01,02,1504
312 DATA 6D00,01,02,02,00,00,02,00,00,01,D [3121]
E,09,00,80,02,00,02,7A2A
313 DATA 6D10,02,00,00,DE,0E,02,00,00,02,0 [3378]
2,80,01,DE,0F,00,80,8FD0
314 DATA 6D20,DE,15,02,03,07,03,03,00,01,D [2882]
E,12,02,00,00,02,80,378E
315 DATA 6D30,00,00,01,DE,06,00,02,DE,05,8 [1942]
0,02,00,00,02,01,00,08C7
316 DATA 6D40,02,00,01,00,00,02,00,00,02,D [4257]
E,05,80,02,00,00,02,1633
317 DATA 6D50,DE,05,00,01,00,00,00,01,02,0 [2222]
2,00,00,DE,06,02,00,5602
318 DATA 6D60,00,00,DE,04,02,00,01,00,02,D [2361]
E,04,00,02,00,00,02,A409
319 DATA 6D70,02,00,00,01,02,80,81,00,00,0 [2304]
1,00,02,DE,04,00,02,0480
320 DATA 6D80,00,80,02,02,80,00,00,DE,05,0 [2422]
2,DE,04,00,02,00,00,A7AC
321 DATA 6D90,02,01,00,DE,07,02,80,80,02,0 [3131]
0,00,01,DE,04,00,02,17FF
322 DATA 6DA0,00,01,02,02,DE,04,00,02,DE,0 [2738]
5,00,02,01,00,00,02,AC6C
323 DATA 6DB0,02,00,00,02,02,01,00,01,00,0 [2787]
0,00,01,00,01,00,01,3514
324 DATA 6DC0,DE,07,00,02,02,80,DE,08,02,D [3168]
E,05,00,DE,08,02,DE,7447
325 DATA 6DD0,06,00,DE,07,02,DE,04,00,1D,0 [3291]
D,FF,FF,00,01,DE,15,CC66
326 DATA 6DE0,02,80,DE,05,00,01,00,00,00,0 [4067]
1,00,00,81,00,00,00,9072
327 DATA 6DF0,80,02,02,01,00,00,00,01,DE,0 [2695]
4,00,81,00,00,01,DE,B78C
328 DATA 6E00,05,00,02,02,00,00,DE,0E,02,0 [2492]
0,00,02,02,00,00,02,5331
329 DATA 6E10,DE,04,00,80,02,02,DE,05,00,0 [4070]
2,00,00,02,02,00,00,A1FF
330 DATA 6E20,02,00,01,00,00,00,02,02,DE,0 [3855]
5,00,02,01,00,02,02,486F
331 DATA 6E30,00,00,02,00,00,02,00,00,02,0 [2255]
2,00,01,02,80,80,02,4A82
332 DATA 6E40,00,00,02,02,00,00,02,00,00,0 [2118]
2,00,00,02,02,00,00,6498
333 DATA 6E50,02,80,80,02,00,00,02,02,00,0 [3711]
0,DE,04,02,00,00,02,3DA3
334 DATA 6E60,02,00,00,DE,04,02,00,00,02,0 [3374]
2,00,00,00,80,DE,04,CA34
335 DATA 6E70,00,02,02,00,00,00,01,DE,04,0 [3347]
0,02,02,80,DE,07,00,1916
336 DATA 6E80,02,02,DE,08,00,DE,15,02,09,0 [4347]
F,11,15,00,01,DE,15,9584
337 DATA 6E90,02,DE,09,00,01,00,00,00,01,0 [3126]
0,80,80,81,02,02,DE,B46D
```

Listing LETSWORK

```
338 DATA 6EA0,06,00,01,00,01,00,00,00,01,0 [3147]
0,00,80,81,80,02,02,268D
339 DATA 6EB0,00,DE,0A,02,00,DE,04,02,00,0 [3228]
0,02,02,DE,12,00,02,94EF
340 DATA 6EC0,02,00,00,00,01,00,01,DE,0C,0 [2994]
0,02,02,DE,04,00,DE,D45F
341 DATA 6ED0,04,02,80,DE,04,02,80,DE,06,0 [3808]
2,DE,12,00,02,02,DE,8FAC
342 DATA 6EE0,06,00,01,00,01,00,01,00,01,0 [3692]
0,01,00,00,00,DE,12,2B0D
343 DATA 6EF0,02,01,00,DE,04,02,DE,07,80,8 [2656]
1,DE,08,00,DE,15,02,4B5D
344 DATA 6F00,0D,03,03,03,00,01,DE,15,02,8 [2539]
0,80,DE,10,00,02,02,8161
345 DATA 6F10,00,DE,04,02,00,00,02,02,02,D [3469]
E,08,00,02,02,00,DE,1171
346 DATA 6F20,04,02,01,00,02,02,02,00,00,D [3028]
E,04,02,00,00,02,02,8B24
347 DATA 6F30,00,00,00,02,02,00,00,02,02,0 [2276]
2,00,00,02,02,00,01,3399
348 DATA 6F40,00,00,02,02,00,02,00,02,02,D [3812]
E,07,00,02,02,00,02,5C7A
349 DATA 6F50,00,00,02,02,80,81,00,02,02,D [3211]
E,04,00,01,00,00,02,500C
350 DATA 6F60,02,00,01,00,00,02,02,00,02,0 [3124]
0,02,02,00,00,DE,04,2CD9
351 DATA 6F70,02,00,02,02,00,02,00,00,02,0 [2678]
2,00,01,00,02,02,00,699D
352 DATA 6F80,01,DE,04,02,80,02,02,DE,04,0 [4395]
0,02,02,00,DE,04,02,7321
353 DATA 6F90,00,00,DE,04,02,00,DE,04,02,0 [2437]
0,00,02,02,80,80,DE,2AF4
354 DATA 6FA0,10,00,DE,15,02,0F,0B,FF,FF,0 [3706]
0,01,DE,16,02,00,00,27EA
355 DATA 6FB0,01,00,80,02,02,02,DE,09,00,0 [4052]
2,02,02,01,02,02,00,0DF5
356 DATA 6FC0,02,02,00,00,02,00,00,01,00,0 [2950]
1,00,00,00,02,02,00,914D
357 DATA 6FD0,01,00,00,01,00,00,01,DE,04,0 [3377]
0,02,02,02,00,00,00,4E70
358 DATA 6FE0,02,02,02,00,02,02,00,02,02,0 [3667]
0,02,00,02,01,00,00,DB55
359 DATA 6FF0,02,80,00,00,02,02,02,00,00,0 [3453]
0,02,80,02,02,02,01,147C
360 DATA 7000,00,00,00,02,80,00,00,00,02,0 [3292]
2,00,01,01,00,02,00,2198
361 DATA 7010,00,80,02,00,00,00,02,80,00,0 [3713]
0,00,02,02,02,00,00,C418
362 DATA 7020,00,02,00,80,80,00,00,02,00,0 [2333]
2,80,00,00,00,DE,04,9334
363 DATA 7030,02,00,00,02,02,00,00,02,00,0 [2356]
1,00,01,00,02,80,00,3359
364 DATA 7040,00,00,02,02,02,00,01,DE,05,0 [2765]
0,02,00,00,00,01,00,AEC2
365 DATA 7050,02,80,00,00,00,02,02,00,00,0 [2509]
2,00,80,00,02,00,00,04A9
366 DATA 7060,01,DE,04,00,02,80,00,00,DE,1 [3728]
5,02,05,03,FF,FF,00,F87F
367 DATA 7070,01,DE,15,02,DE,12,00,02,02,0 [3072]
0,00,01,00,00,01,DE,3BFF
368 DATA 7080,0C,00,02,02,00,02,01,02,01,D [2463]
E,04,02,00,01,02,DE,5F78
369 DATA 7090,06,00,02,02,00,02,00,00,01,0 [2416]
0,00,00,02,00,00,02,6891
370 DATA 70A0,DE,06,00,02,02,00,02,DE,06,0 [2768]
0,02,00,00,02,DE,06,689C
371 DATA 70B0,00,02,02,00,DE,04,02,01,00,0 [2376]
1,00,01,00,02,DE,06,24E4
372 DATA 70C0,00,02,02,DE,05,00,01,00,00,0 [2960]
2,01,00,02,00,00,DE,0A63
373 DATA 70D0,04,80,02,02,DE,05,00,02,01,0 [2783]
0,01,00,01,02,00,00,8684
374 DATA 70E0,DE,04,80,02,02,DE,05,00,02,0 [3429]
0,01,00,02,00,02,00,4349
375 DATA 70F0,00,DE,04,80,02,02,DE,0A,00,0 [2580]
1,00,00,00,DE,04,80,AD8E
376 DATA 7100,DE,15,02,05,11,FF,FF,00,01,0 [2099]
0,00,DE,12,02,00,00,D790
377 DATA 7110,02,DE,07,00,01,00,81,DE,04,8 [2095]
0,02,00,02,00,00,02,9665
378 DATA 7120,00,02,02,01,02,01,02,02,00,D [1881]
E,06,02,00,02,00,00,F568
```

Listing LETSWORK

```
379 DATA 7130,02,00,02,DE,05,00,02,00,00,0 [3426]
1,00,00,02,80,80,02,8F5E
380 DATA 7140,00,00,02,00,02,DE,05,00,02,0 [2330]
1,00,00,01,00,02,00,234F
381 DATA 7150,00,02,00,00,02,00,DE,04,02,0 [2831]
0,02,02,00,01,01,00,2967
382 DATA 7160,00,02,00,00,02,00,00,02,DE,0 [4122]
5,00,01,00,01,DE,05,FDED
383 DATA 7170,00,02,00,00,DE,09,02,80,00,0 [1563]
0,00,DE,05,02,00,00,DDC6
384 DATA 7180,02,02,80,80,DE,04,00,DE,04,0 [3403]
2,DE,08,00,02,02,80,A753
385 DATA 7190,80,DE,07,00,01,01,01,DE,04,0 [3319]
2,00,00,02,02,80,80,B36F
386 DATA 71A0,DE,10,00,DE,15,02,17,0D,FF,F [2949]
F,00,01,DE,16,02,DE,2544
387 DATA 71B0,05,00,02,02,DE,0A,00,02,02,D [3378]
E,04,00,02,01,00,01,0C11
388 DATA 71C0,DE,05,00,02,00,02,02,00,02,0 [2748]
2,DE,04,00,02,00,02,7664
389 DATA 71D0,00,DE,05,02,00,00,00,02,00,0 [3315]
2,02,80,80,80,00,00,0CF7
390 DATA 71E0,00,01,00,00,00,01,00,00,01,0 [3337]
0,00,02,00,02,02,80,442C
391 DATA 71F0,80,80,00,02,01,02,00,02,01,0 [3535]
0,02,02,00,02,00,02,228A
392 DATA 7200,00,02,02,80,80,80,00,02,00,0 [2668]
2,01,02,00,02,02,00,C282
393 DATA 7210,01,00,00,01,00,02,02,80,80,8 [2608]
0,DE,05,00,01,00,02,6796
394 DATA 7220,02,00,02,02,00,02,00,02,02,D [2581]
E,04,00,02,01,02,02,5C13
395 DATA 7230,02,00,00,02,00,02,00,00,02,0 [2367]
0,02,02,DE,12,00,DE,2F07
396 DATA 7240,04,02,DE,04,00,02,02,00,00,0 [3215]
0,02,00,02,DE,04,00,0F39
397 DATA 7250,DE,15,02,17,05,FF,FF,00,00,0 [1867]
0,00,00,00,00,00,00,5A69
398 DATA *ENDE*                             [502]
399 adr=&6000:zeile=104:MEMORY adr-1       [2251]
400 READ d$:IF d$="*ENDE*"THEN 411         [1165]
401 pr=0                                    [117]
402 FOR i=1 TO 16                           [317]
403 READ a$:a=VAL("&"+a$)                   [883]
404 POKE adr,a:adr=adr+1                    [717]
405 pr=pr*2:IF pr>65535 THEN pr=pr-65535    [2485]
406 pr=UNT(pr)XOR a:IF pr<0 THEN pr=pr+655 [1890]
36
407 NEXT i                                  [375]
408 READ pr$:pr2=VAL("&"+pr$):IF pr2<0 THE [2610]
N pr2=pr2+65536
409 IF pr<>pr2 THEN PRINT"Pruefsummenfehle [3615]
r in Zeile";zeile:STOP
410 zeile=zeile+1:GOTO 400                  [996]
411 SAVE"COMPLEVE.BIN",B,&6000,&1258        [1628]
412 PRINT d$:END                            [346]
```

```
10 REM                                      [272]
20 REM Entcomp                              [812]
30 REM                                      [272]
40 MEMORY &3FFF:LOAD"compleve.bin",&6000:q [2989]
adr=&6000:zadr=&4000
50 IF zadr>&5CD3 THEN 80                    [1090]
60 w=PEEK(qadr):IF w<>222 THEN POKE zadr,w [3973]
:zadr=zadr+1:qadr=qadr+1:GOTO 50
70 FOR i=1 TO PEEK(qadr+1):POKE zadr,PEEK( [3204]
qadr+2):zadr=zadr+1:NEXT:qadr=qadr+3:GOTO
50
80 PRINT"Bitte Diskette bzw. Kassette mit  [10514]
LETSWORK.BIN einlegen und Taste druecken !
":CALL &BB18
90 SAVE"level.let",b,&4000,&1CD4            [1407]
```

Listing LETSWORK

BISTRO 'LA CUISINE'

Der CPC als Küchenchef

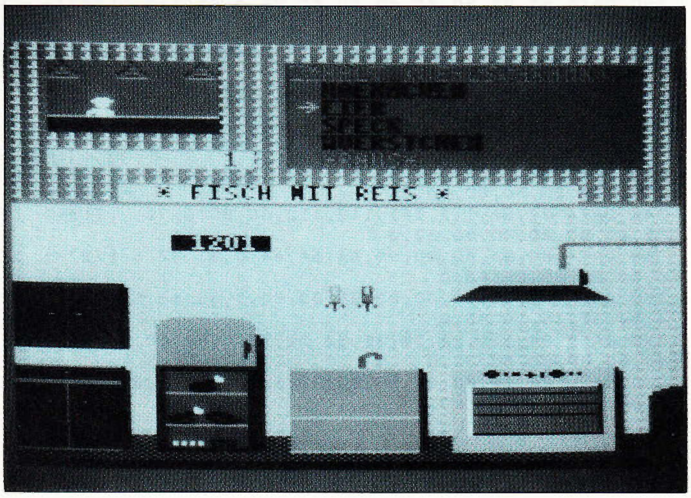
Das Programm ist nicht etwa eine Verwaltung für Mutters Küche, sondern ein Spielprogramm, in dem Joystick-Akrobaten ihr Geschick im Zubereiten von Mahlzeiten beweisen können, ohne daß gleich die ganze Wohnung abbrennt.

Sie sind als Koch in einem beliebten Bistro angestellt und müssen sich dem Andrang der Gäste durch exaktes und schnelles Zubereiten der bestellten Gerichte stellen.

Nach dem Starten erscheint das Titelbild, in dem Sie Paul Combuse, der oberste aller Computer-Köche, begrüßt. Durch Drücken der Feuertaste oder COPY kann nun der Schwierigkeitsgrad des Spiels gewählt werden, von dem auch die Spiellänge abhängt. Danach wird das Spielscreen aufgebaut. Dieser ist in zwei Hauptteile geteilt, in der oberen Hälfte werden Mitteilungen ausgegeben, während sich in dem unteren Teil der Aktionsbildschirm präsentiert.

Die Aufgabe des Spielers ist es nun, mit dem Cursor die jeweiligen Schränke sowie die benötigten Küchengeräte anzusteuern und das bestellte Gericht zuzubereiten. Dem Computer sind verschiedene Gerichte bekannt, deren Rezepte der Spieler im Hauptmenü, welches während der Cursorsteuerung eingeblendet ist, abfragen kann. Das aktuelle Rezept kann aufgerufen werden, aber auch das Ausdrucken einer Gesamtliste auf Bildschirm oder Drucker ist möglich. Durch das Rezept erfährt man nun, welche Zutaten benötigt werden und wie sie weiter zubereitet werden müssen. Die angegebenen Zubereitungsschritte (zum Beispiel Bratzeit, Schneiden, Würzen und so weiter) müssen genaustens und in der aufgelisteten Reihenfolge durchgeführt werden. Jede Abweichung von der Rezeptvorgabe führt unweigerlich zur Verärgerung des Gastes und somit zu Punktabzügen. Des weiteren läßt sich eine Inventarangabe abrufen.

Es können zwei Inventarteile aufgenommen werden (Auch ein Computer-Koch hat nur zwei Hände!). Jeweils eine Zutat und ein Küchengegenstand. Dadurch ist volle Konzentration nötig, denn die Aufnahme eines falschen oder ungewünschten Teils kostet wichtige Zeit und das Gericht muß in den Mülleimer wandern. Hat man es nun geschafft, ein bestelltes Gericht zusammenzustellen und zu plazieren, kann man es mit dem Menüpunkt 'Bestellung' fertig aus der Küche herausgeben. Danach gibt der Gast sein Urteil für das Gericht ab, und je nach Vollständigkeit äußert er sich. Verärgerung oder Zufriedenheit schlagen sich dann später auch in der Auswertung nieder. Die Spieldauer hängt vom jeweils gewählten Schwierigkeitsgrad ab: Bei Level 1 von 7:30 bis 8:00, bei Level 2 von 12:00 bis 12:45 und bei Level 3 von 19:00 bis 20:00. Achtung, keine Echtzeit!!! Nach der jeweiligen Spielzeit bricht das Programm ab und eine Endauswertung Ihrer erreichten Punktzahl sowie ein Urteil über Ihre Fähigkeiten als Koch werden ausgegeben. Natürlich ist nicht alles sofort zu finden beziehungsweise zu verstehen. Da hilft eben nur ausprobieren. Wenn man einmal den Spielablauf 'drin' hat, macht das Spiel riesigen Spaß und kann einen für Stunden an den Bildschirm fesseln.



Durch die grafische Untermalung kann man sich auch als Anfänger in der Küche gut zurechtfinden

Um das Finden von Zutaten und anderem zu erleichtern, steht hier nun...

Der Aufbau des Spielscreens

AUSGABE-FENSTER: OBEN RECHTS
Texte und Meldungen zu den Menüpunkten 'Kochrezepte', 'Inventar' und 'Bestellungen' werden hier ausgegeben. Ebenso werden hier die Inhalte der Schränke und die Funktionen der Küchengeräte angezeigt und können angewählt werden.

BISTRO-FENSTER: OBEN LINKS
Jeder Gast, der das Bistro betritt und eine Bestellung aufgibt, erscheint in diesem Fenster.

BESTELLUNGEN-ANZEIGE: OBEN LINKS
Die Anzahl der bisher eingegangenen Bestellungen wird angezeigt.

BESTELLUNGEN-ANZEIGE: MITTE
Die Bezeichnung für das bestellte Gericht erscheint hier, und das jeweilige Kochrezept kann nun abgefragt werden.

KÜCHENUHR: MITTE LINKS
Mit ihr läßt sich die Brat-, Koch- oder Backzeit überprüfen. Keine Echtzeit!

WANDSCHRANK: UNTERE HÄLFTE LINKS
Hier finden Sie Teller und Suppenschüsseln.

MEHRZWECKFACH 1: UNTEN LINKS
Suppenextrakt für die Suppenzubereitung und Fertigsoßen befinden sich hier.

BODENSCHRANK: UNTEN LINKS
Hier liegen Nudeln, Reis und Brötchen.

GEFRIERFACH: UNTEN LINKS/MITTE
Pizza, Baguette, Hamburger und Fischfilet werden hier aufbewahrt.

KÜHLSCHRANK: UNTEN LINKS/MITTE
Hier kann man sich Hähnchen, Eier, Speck und Würstchen holen.

HÄNGEFLASCHEN: UNTEN MITTE
Aus den Spendern, die sich an der Wand befinden, kann wahlweise Senf oder Ketchup entnommen werden.

ABLAGE: UNTEN MITTE
Auf der Arbeitsfläche, die sich direkt neben dem Spülbecken befindet, werden die 'manuellen' Arbeitsvorgänge Schneiden, Würzen und Plazieren ausgeführt. Auch kann die Ablage von Teller und Zutaten gereinigt werden; falls etwas Falsches plaziert wurde.

SPÜLE: UNTEN MITTE
Hier müssen Reis, Fischfilet und Hähnchen vor der Weiterverarbeitung gewaschen werden.

BEHÄLTERDEPOT: UNTEN MITTE
Töpfe und Pfannen, die zum Kochen beziehungsweise Braten benutzt werden, sind hier.

MEHRZWECKFACH 2: UNTERE HÄLFTE/OBEN RECHTS
Gewürze, die zum Würzen der Zutaten benutzt werden, und ein Küchenmesser sind hier aufbewahrt.

DUNSTABZUG: UNTERE HÄLFTE/ OBEN RECHTS
Der Dunstabzug wird automatisch beim Betrieb des Herdes eingeschaltet, hat aber keine Funktion für den Spieler!

HERD: UNTEN RECHTS
Der Herd hat die Funktionen 'Braten', 'Kochen' und 'Backen'. Mit einer Übersichtskarte können die Herdplatten und der Backofen überprüft werden.

MÜLLEIMER: UNTEN RECHTS
Er ist die Notlösung für falsche oder verbrannte Zutaten. Das gesamte Inventar wird ausgeleert.

(Thorsten Mertsching/jb)

Folgende Abkürzungen werden bei der Herdübersicht benutzt:

BR – Braten (wenn gewählt)
KO – Kochen
BA – Backen
06 – *Zeit bis Zutat fertig (hier 6 'Minuten')*
OK – *Zutat fertig und von Herd nehmen*
VB – *Zutat verbrannt und muß in Mülleimer*
P1, P2, ... – Anzeige der Herdplatten (1-4)
B5 – Backofen (Menünummer 5)

für 464-664-6128

```
1 REM La Cuisine                               [957]
2 REM Datalader fuer Cuisine.ste               [2560]
10 adr=&8FFF:FOR i=1 TO 12:FOR j=1 TO 60:R [6386]
EAD b$:a=VAL("&"+b$):POKE adr+i*j,a:b=b+a:
NEXT:PRINT 20+i*10:NEXT
20 IF b=50078 THEN SAVE"CUISINE.STE",b,&90 [4654]
00,&DF ELSE PRINT"ERROR!":STOP
30 DATA 21,00,F8,DD,7E,00,3D,FE,00,28,07,0 [8984]
1,50,00,09,3D,20,FC,16,00,DD,5E,02,1B,19,C
9,FE,02,C0,CD,00,90,11,DE,90,06,04,7E,12,1
3,23,7E,12,23,13,7E,12,13,23,7E,12,CD,29,B
C,13,7E,12,2B,13,7E
40 DATA 00,DD,00,FE,28,01,00,3D,FC,00,5E,1 [9102]
B,C9,02,CD,90,DE,06,7E,13,7E,23,7E,13,7E,C
D,BC,7E,2B,7E,2B,7E,2B,7E,CD,BC,10,C9,02,C
D,90,BE,06,CD,90,CD,90,CD,90,CD,90,29,CD,9
0,CD,90,CD,90,CD,90
50 DATA F8,00,00,01,09,FC,DD,1B,FE,CD,11,0 [9391]
6,12,7E,13,13,12,BC,12,7E,13,2B,12,BC,DA,0
2,00,BE,04,90,7E,CD,23,90,BC,90,7E,CD,2B,9
0,BC,C9,13,BD,A1,C1,02,00,DE,04,13,77,1A,1
3,CD,13,2B,77,1A,2B
60 DATA DD,FE,01,3D,00,1B,02,90,06,13,23,1 [9851]
3,CD,7E,7E,7E,7E,BC,C9,CD,BE,CD,CD,CD,CD,2
9,90,90,90,90,10,1A,BD,AE,FE,00,90,77,77,7
7,77,13,13,13,2B,29,DA,55,AA,55,FF,FF,AA,5
5,AA,00,00,00,00,00
70 DATA 7E,28,09,00,19,CD,90,13,13,7E,13,7 [9686]
E,12,CD,DA,CD,90,90,23,CD,BC,CD,7E,90,DA,4
F,A1,FE,90,04,1A,77,CD,77,2B,2B,BC,C0,00,5
5,FF,FF,00,AA,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00
80 DATA 00,01,FC,1B,CD,06,7E,13,BC,7E,2B,B [9784]
C,02,BE,90,CD,90,90,CD,90,C9,BD,C1,00,04,7
7,13,13,77,2B,13,55,00,FF,FF,55,00,00,00,0
0,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00
90 DATA 3D,00,DD,02,90,7E,23,7E,13,CD,FE,B [9685]
E,23,90,BC,90,7E,C9,90,FE,DE,23,23,13,2B,7
7,C9,AA,FF,FF,AA,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00,00,00,00,00,00,00,00,0
0,00,00,00,00,00,00
100 DATA FE,3D,1B,90,13,13,7E,7E,BC,CD,CD, [7227]
CD,29,90,90,1A,AE,00,77,77,13,13,29,55,55,
FF,55,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00
110 DATA 00,FC,FE,06,13,BC,13,BC,00,90,23, [7689]
90,2B,C9,A1,00,13,13,2B,2B,C9,00,FF,55,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00
120 DATA 28,00,CD,13,7E,7E,CD,CD,90,CD,CD, [9459]
90,4F,FE,04,77,77,2B,C0,55,FF,AA,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00
130 DATA 07,5E,11,23,13,2B,FE,CD,23,CD,CD, [10307]
BD,CD,23,CD,13,10,00,FF,AA,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00
140 DATA 01,1B,06,13,7E,BC,BE,CD,90,90,BD, [6739]
00,77,13,2B,55,FF,55,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00,00,00,00,00,00,00,00,
00,00,00,00,00,00,00
```

Listing CUISINE

```
10 ' *************************             [768]
20 ' *   L A   C U I S I N E   *             [311]
30 ' * ----------------------- *             [1284]
40 ' *     ZEICHENSATZLADER    *             [1121]
50 ' *************************             [768]
60 '                                          [117]
70 MODE 1:BORDER 0:INK 0,0:INK 1,26:INK 2, [2882]
12:INK 3,6:PAPER 2:PEN 1
80 SYMBOL AFTER 32:WINDOW#1,8,33,5,16:PAPE [5595]
R#1,3:CLS#1:PEN#1,1:LOCATE#1,3,2
90 FOR i=1 TO 3:a=i*2:PLOT 110-a,142-a,i:D [3625]
RAW 110-a,337+a:DRAW 528+a,337+a
100 DRAW 528+a,142-a:DRAW 110-a,142-a:NEXT [4179]
:PRINT#1,"Herzlich Willkommen zu"
110 PEN#1,2:LOCATE#1,8,5:PRINT#1,"*LA CUIS [3470]
INE*":LOCATE#1,3,9:PEN#1,0
120 PRINT#1,"Programm & Grafik 1988":LOCAT [4578]
E#1,2,11
130 PRINT#1,"von  Thorsten Mertsching":s=6 [4791]
00:EVERY 40,1 GOSUB 180
140 LOCATE 14,22:PRINT"LADEVORGANG...":FOR [3619]
 i=1 TO 75:FOR j=1 TO 9:READ a(j)
150 sum=sum+a(j):NEXT:SYMBOL a(1),a(2),a(3 [4453]
),a(4),a(5),a(6),a(7),a(8),a(9)
160 NEXT:PEN 1:IF sum<>51087 THEN CLS:PRIN [4789]
T"FEHLER IN DATA!":STOP
170 a=REMAIN(1):MEMORY &8FFF:LOAD"!CUISINE [6732]
.STE":RUN"!CUISINE.PRG"
180 s=s-10:SOUND 1,s,0,7:SOUND 2,s+2,0,7:R [2538]
ETURN
190 DATA 33,32,32,32,32,32,0,32,0,39,32,32 [2958]
,64,0,0,0,0,0
200 DATA 44,0,0,0,0,0,32,32,64,63,32,80,16 [2733]
,32,32,0,32,0,46,0,0,0,0,0,32,0
210 DATA 42,0,80,32,112,32,80,0,0,45,0,0,0 [2868]
,112,0,0,0,0
220 DATA 48,32,80,80,80,80,80,32,0,49,32,9 [2831]
6,32,32,32,32,112,0
230 DATA 50,32,80,16,32,32,64,112,0,51,32, [2433]
80,16,32,16,80,32,0
240 DATA 52,64,64,80,112,16,16,16,0,53,112 [2411]
,64,64,96,16,80,32,0
250 DATA 54,32,80,64,96,80,80,32,0,55,112, [1963]
16,16,32,32,64,64,0
260 DATA 56,32,80,80,32,80,80,32,0,57,32,8 [2103]
0,80,48,16,80,32,0
270 DATA 58,0,0,32,0,32,0,0,0,65,32,80,80, [1650]
112,80,80,80,0
280 DATA 66,96,80,80,96,80,80,96,0,67,32,8 [2862]
0,64,64,64,80,32,0
290 DATA 68,96,80,80,80,80,80,96,0,69,112, [2631]
64,64,96,64,64,112,0
300 DATA 70,112,64,64,96,64,64,64,0,71,32, [2543]
80,64,112,80,80,32,0
310 DATA 72,80,80,80,112,80,80,80,0,73,112 [3194]
,32,32,32,32,32,112,0
320 DATA 74,112,16,16,16,16,80,32,0,75,80, [2561]
80,96,96,96,80,80,0
330 DATA 76,64,64,64,64,64,64,112,0,78,80, [2627]
112,112,80,80,80,80,0
340 DATA 77,80,112,112,112,112,80,80,0,79, [3135]
32,80,80,80,80,80,32,0
350 DATA 80,96,80,80,96,64,64,64,0,81,32,8 [2275]
0,80,80,80,96,48,0
360 DATA 82,96,80,80,96,96,80,80,0,83,48,6 [2886]
4,64,32,16,16,96,0
370 DATA 84,112,32,32,32,32,32,32,0,85,80, [2801]
80,80,80,80,80,32,0
380 DATA 86,80,80,80,80,80,32,32,0,87,80,8 [2566]
0,80,80,112,112,80,0
```

Listing CUISINE

```
390 DATA 88,80,80,112,32,112,80,80,0,89,80 [3081]
,80,80,80,32,32,32,0
400 DATA 90,112,16,16,32,64,64,112,0,165,2 [3607]
55,153,153,255,255,153,153,255
410 DATA 166,68,68,204,0,68,68,204,0,167,1 [2483]
6,16,56,40,108,198,254,0
420 DATA 168,28,62,28,28,8,60,126,94,169,3 [3633]
1,127,127,255,255,255,255,255
430 DATA 170,248,254,254,255,255,255,255,2 [3947]
55,171,192,192,192,192,192,192,192,192
440 DATA 172,128,128,192,192,192,192,192,1 [3914]
92,173,192,192,192,192,192,192,128,128
450 DATA 174,0,4,2,31,2,4,0,0,175,28,62,11 [3317]
8,102,96,96,96,96
460 DATA 176,56,84,84,16,16,0,0,0,177,56,4 [1623]
0,40,40,56,56,56,56
470 DATA 178,52,126,126,62,60,60,60,60,179 [2925]
,60,60,90,126,126,90,36,60
480 DATA 180,60,126,247,189,181,255,118,60 [2906]
,181,0,96,245,245,96,0,0,0
490 DATA 182,0,0,146,186,18,0,0,0,183,0,25 [2827]
5,126,255,126,255,126,60
500 DATA 184,0,0,62,62,28,62,62,28,185,0,0 [2829]
,12,124,248,252,252,120
510 DATA 186,2,3,3,0,0,0,0,0,187,0,126,70, [2235]
70,114,126,78,126
520 DATA 188,0,0,56,56,12,0,48,0,189,0,0,0 [2347]
,0,24,126,255,255
530 DATA 190,0,0,0,0,255,127,127,63,191,0, [1891]
0,7,31,248,240,240,224
540 DATA 192,0,0,0,0,0,85,85,0,193,0,0,112 [2732]
,248,220,207,127,60
550 DATA 194,0,0,0,0,126,90,86,126,195,0,0 [2864]
,7,31,127,127,255,255
```

```
10 ' ************************             [768]
20 ' *   L A   C U I S I N E   *           [311]
30 ' * -------------------- *              [1284]
40 ' *      HAUPTPROGRAMM      *           [1465]
50 ' ************************             [768]
60 '                                       [117]
70 PAPER 0:FOR i=1 TO 25:LOCATE 40,25:PRIN [2607]
T" ":NEXT
80 ENV 1,=9,2000:ENV 2,2,15,1,7,-1,2,15,-1 [1338]
,4
90 agr=10:itu=1:schr(1)=2:schr(2)=2:RANDOM [3726]
IZE TIME*4
100 meld$(1)="MIT WAS DENN?":meld$(2)="WOR [4766]
AUF DENN?":meld$(3)="NICHT NOETIG!"
110 DIM inf$(15,8),f(15,20),xcol(17),nr(13 [3719]
),bz$(13),zub$(10,11),df$(10,11)
120 DIM ob(13),un(13),li(13),re(13),ina(13 [3991]
),in$(13,4),wt(13,4),zt(9,4)
130 zt(2,1)=2:zt(2,2)=1:zt(3,1)=4:zt(3,2)= [4712]
4:zt(6,1)=4:zt(6,2)=2
140 zt(6,3)=1:zt(6,4)=3:zt(7,1)=4:zt(7,2)= [4360]
4:zt(7,3)=3:zt(7,4)=4
150 :                                      [174]
160 ' *** Titelbild ***                    [600]
170 :                                      [174]
180 MODE 0:INK 0,0:BORDER 2:PAPER 0:snd=-1 [3161]
190 RESTORE 5760:FOR i=1 TO 15:READ col:IN [3174]
K i,col:NEXT
200 x=3:y=7:GOSUB 5440:x=18:GOSUB 5440:FOR [3637]
 i=0 TO 4 STEP 2:j=i/2+1
210 PLOT i,i,11+j:DRAW i,399-i:DRAW 639-i, [4040]
399-i:DRAW 639-i,i:DRAW i,i:NEXT
220 a$="THOM SOFTWARE PRAESENTIERT":xw=4:y [3009]
w=3:p=12:GOSUB 5340
230 FOR i=0 TO 6:READ urt$(i):NEXT:LOCATE  [3213]
6,8:PEN 10:PRINT"LA CUISINE"
240 LOCATE 5,9:PEN 5:PRINT STRING$(11,154) [3327]
CHR$(152):i=1
250 a$="COPYRIGHT '89 BY":xw=7:yw=13:p=13: [3696]
GOSUB 5340
260 a$="PC AMSTRAD INTERNATIONAL":xw=5:yw= [4402]
15:GOSUB 5340
```

Listing CUISINE

```
270 FOR i=1 TO 13:nr(i)=i:READ bz$(i),ob(i [3837]
),un(i),li(i),re(i)
280 FOR j=ob(i) TO un(i):FOR k=li(i) TO re [4485]
(i):f(j-10,k)=nr(i):NEXT k,j
290 READ ina(i):FOR j=1 TO ina(i):READ in$ [3013]
(i,j):NEXT j,i
300 FOR i=1 TO agr:READ rp$(i),zbs(i):FOR  [6108]
j=1 TO zbs(i):READ zub$(i,j):NEXT j,i
310 a$="DRUECKE COPY FUER KEYBOARD SPIEL": [4492]
xw=3:yw=20:p=9:GOSUB 5340
320 a$="ODER FEUERKNOPF FUER JOYSTICK SPIE [3488]
L":xw=2:yw=22:GOSUB 5340
330 IF i/2=INT(i/2) THEN sd=5:a$=" ":b$=CH [4549]
R$(178) ELSE a$=CHR$(178):b$=" ":sd=1
340 INK 10,RND*27:i=i+1:PEN 1:LOCATE 3,6:P [3052]
RINT a$:LOCATE 18,6:PRINT b$
350 LOCATE 3,7:PRINT b$:LOCATE 18,7:PRINT  [2105]
a$
360 SOUND 7,1000/sd,25,,2,,3:FOR p=1 TO 40 [2295]
370 IF NOT INKEY(9) THEN lin=8:rec=1:vor=0 [2992]
:hin=2:feu=9:GOTO 390
380 IF NOT INKEY(76) THEN lin=74:rec=75:vo [3055]
r=72:hin=73:feu=76 ELSE NEXT:GOTO 330
390 INK 10,23:FOR i=90 TO 40 STEP -5:SOUND [1430]
 1,i,1,15:NEXT
400 WINDOW#1,2,19,20,24:PAPER#1,3:CLS#1:PL [3695]
OT 30,15,5
410 DRAW 30,97:DRAW 610,97:DRAW 610,15:DRA [1590]
W 30,15
420 a$="** WAEHLE  SCHWIERIGKEITSGRAD **": [3643]
xw=3:yw=18:p=12:GOSUB 5340
430 a$="1 --- LEVEL 1: LEICHT":xw=5:yw=21: [3083]
p=13:GOSUB 5340
440 a$="2 --- LEVEL 2: SCHWIERIG":yw=22:p= [2726]
14:GOSUB 5340
450 a$="3 --- LEVEL 3: HART":yw=23:p=15:GO [3094]
SUB 5340
460 a$=INKEY$:IF a$<"1" OR a$>"3" THEN 460 [2452]
 ELSE lev=VAL(a$)
470 SOUND 7,lev*80,50,,1:IF lev=1 THEN st= [4968]
7:mi=30:el=8:e2=0
480 IF lev=2 THEN st=12:mi=0:el=12:e2=45   [1430]
490 IF lev=3 THEN st=19:mi=0:el=20:e2=0    [1284]
500 OUT &BC00,1:FOR i=40 TO 0 STEP -1:OUT  [4563]
&BD00,i:FOR j=1 TO 20:NEXT j,i
510 :                                      [174]
520 ' *** Pausenmelodie ***                [933]
530 :                                      [174]
540 RESTORE 590:EVERY 10,1 GOSUB 550:GOTO  [1435]
4100
550 a1=SQ(1):a2=SQ(2):IF a1>0 AND a2>0 THE [2629]
N GOSUB 570
560 RETURN                                 [555]
570 READ ton:IF ton=999 THEN RESTORE 590:G [2791]
OTO 570
580 SOUND 2,ton,20,,1:SOUND 1,ton,1,,4,,10 [4598]
:RETURN
590 DATA 239,179,179,142,159,179,159,142,1 [2346]
79,119,106,119,179
600 DATA 159,159,142,179,119,239,179,999   [1524]
610 :                                      [174]
620 ' *** Steuerung ***                    [1376]
630 :                                      [174]
640 xp=(x-1)*4+1:yp=y:IF NOT nix THEN CALL [4565]
 &901A,xp,yp:CALL &904C,xp,yp
650 IF NOT INKEY(lin) AND xp>1 THEN CALL & [3411]
908B,xp,yp:xp=xp-4:GOTO 740
660 IF NOT INKEY(rec) AND xp<77 THEN CALL  [2369]
&908B,xp,yp:xp=xp+4:GOTO 740
670 IF NOT INKEY(vor) AND yp>10 THEN CALL  [3168]
&908B,xp,yp:yp=yp-1:GOTO 740
680 IF NOT INKEY(hin) AND yp<25 THEN CALL  [3111]
&908B,xp,yp:yp=yp+1:GOTO 740
690 IF NOT INKEY(feu) THEN feuer=-1:SOUND  [1978]
2,200,1,15:GOTO 760
700 IF NOT INKEY(37) THEN mnK=-1:GOTO 760  [1876]
710 IF NOT INKEY(35) THEN mnI=-1:GOTO 760  [1696]
720 IF NOT INKEY(54) THEN mnB=-1:GOTO 760  [2029]
730 GOTO 650                               [371]
740 CALL &901A,xp,yp:CALL &904C,xp,yp      [1640]
750 FOR ip=1 TO 10:NEXT                    [469]
760 y=yp:x=(xp-1)/4+1:xpo=x:ypo=y:nix=0    [4191]
770 IF mnK THEN mnK=0:GOTO 3150            [1577]
780 IF mnI THEN mnI=0:GOTO 3600            [1734]
790 IF mnB THEN mnB=0:GOTO 3690            [1605]
800 IF NOT feuer THEN 650                  [195]
```

Listing CUISINE

```
810 :                                          [174]
820 ' *** Einsicht Kueche ***                  [1743]
830 :                                          [174]
840 feuer=0:IF f(y-10,x)=0 THEN nix=-1:GOT     [3734]
O 640 ELSE f=f(y-10,x)
850 IF f=13 THEN 1250                          [596]
860 IF f=4 OR f=5 OR f>9 THEN 1320             [2004]
870 WINDOW#7,li(f),re(f),ob(f),un(f):PAPER     [3800]
#7,12:CLS#7
880 x1=li(f):x2=re(f):y1=ob(f):y2=un(f)        [2233]
890 DI:xa=x1*32-32:xe=x2*32-2:ya=400-y1*16     [3243]
+16:ye=400-y2*16+4
900 FOR i=1 TO 2:PLOT xa,ya-i*2,7:DRAW xe,     [2678]
ya-i*2
910 PLOT xa,ye-i*2:DRAW xe,ye-i*2:PLOT xa+     [4023]
i*2,ya-2:DRAW xa+i*2,ye
920 PLOT xe-i*2,ya-2:DRAW xe-i*2,ye:NEXT:E     [2814]
I
930 IF f>1 THEN 980                            [809]
940 PLOT 8,148,13:DRAW 84,148:cw=1:PRINT C     [3336]
HR$(23)CHR$(cw);
950 TAG:tg=-1:PLOT -2,-2,13:MOVE 14,130:PR     [3463]
INT CHR$(183);
960 MOVE 48,130:PRINT CHR$(183);:MOVE 14,1     [3825]
64:PRINT CHR$(184);
970 MOVE 48,164:PRINT CHR$(184);:GOTO 1280     [2414]
980 IF f<>3 THEN 1060                          [930]
990 PLOT 10,50:DRAW 86,50,10:PLOT 0,-2,4:P     [3880]
RINT CHR$(23)CHR$(1);
1000 cw=1:TAG:tg=-1:MOVE 15,67:PRINT CHR$(     [3671]
187);:PLOT 0,-2,13:MOVE 15,67
1010 PRINT CHR$(188);:PLOT 0,-2,5:MOVE 42,     [4455]
67:PRINT CHR$(187);:PLOT 0,-2,7
1020 MOVE 42,67:PRINT CHR$(188);:PLOT 0,-2     [4673]
,6:MOVE 8,35:PRINT CHR$(187);
1030 PLOT 0,-2,3:MOVE 8,35:PRINT CHR$(188)     [3099]
;:PLOT 0,-2,5:MOVE 40,35
1040 PRINT CHR$(187);                          [1213]
1050 GOTO 1280                                 [347]
1060 IF f<>6 THEN 1140                         [567]
1070 PLOT 136,70:DRAW 214,70,15:PLOT 136,4     [3620]
0:DRAW 214,40
1080 PLOT 0,-2,11:PRINT CHR$(23)CHR$(1);:c     [2724]
w=1:TAG:tg=-1
1090 MOVE 140,56:PRINT CHR$(185);:MOVE 160     [4094]
,86:PRINT CHR$(185);
1100 PLOT 0,-2,13:MOVE 140,56:PRINT CHR$(1     [3344]
86);:MOVE 160,86:PRINT CHR$(186);
1110 PLOT 0,-2,11:MOVE 176,56:PRINT CHR$(1     [2772]
93);:PLOT 0,-2,13
1120 MOVE 140,34:PRINT CHR$(192);:PLOT 0,-     [3796]
2,4:MOVE 178,34:PRINT CHR$(194);
1130 GOTO 1280                                 [347]
1140 IF f<>7 THEN 1200                         [1132]
1150 PLOT 0,-2,5:cw=1:PRINT CHR$(23)CHR$(1     [2916]
);:TAG:tg=-1
1160 MOVE 145,114:PRINT CHR$(187);:PLOT 0,     [4033]
-2,13:MOVE 145,114:PRINT CHR$(188);
1170 PLOT 0,-2,2:MOVE 177,114:PRINT CHR$(1     [3688]
87);:PLOT 0,-2,10:MOVE 177,114
1180 PRINT CHR$(188);                          [1241]
1190 GOTO 1280                                 [347]
1200 IF f<>8 THEN 1250                         [615]
1210 cw=1:LOCATE 1,1:PRINT CHR$(23)CHR$(1)     [2863]
;:TAG:tg=-1
1220 PLOT 0,-2,8:MOVE 272,50:PRINT CHR$(18     [4283]
9);:PLOT 268,32,5:PLOT 268,30
1230 PLOT 0,-2:MOVE 272,34:PRINT CHR$(143)     [3695]
;:PLOT 304,32:PLOT 304,30
1240 PLOT 0,-2,8:MOVE 310,34:PRINT CHR$(19     [5299]
0);:MOVE 340,34:PRINT CHR$(191);
1250 IF f<>13 THEN 1280                        [1238]
1260 IF itu=1 THEN PRINT CHR$(7);:nix=-1:G     [1727]
OTO 640
1270 LOCATE#1,20,12:PRINT#1,CHR$(22)CHR$(0     [2922]
);CHR$(165);CHR$(22)CHR$(1);
1280 TAGOFF:tg=0:cw=0:LOCATE 1,1:PRINT CHR     [2939]
$(23)CHR$(cw);
1290 :                                         [174]
1300 ' *** Inhalte auflisten ***               [900]
1310 :                                         [174]
1320 IF f>9 THEN 1350                          [688]
1330 PAPER#4,12:CLS#4:p=8:a$="INHALT "+bz$     [3595]
(f)+":"
1340 xw=9:yw=2:GOSUB 5340:xw=10:GOTO 1360      [2871]
1350 PAPER#4,13:CLS#4:p=8:a$="FUNKTIONEN "     [4658]
+bz$(f)+":":xw=9:yw=2:GOSUB 5340
```

Listing CUISINE

```
1360 xw=10:FOR s=1 TO ina(f):a$=in$(f,s):p     [3695]
=10:yw=yw+1:GOSUB 5340
1370 NEXT:yw=yw+1:p=5:a$="*RAUS*":GOSUB 53     [5092]
40:maxi=ina(f)+2
1380 x=xpo:y=ypo:tg=0:TAGOFF:cw=0:PRINT CH     [3161]
R$(23)CHR$(0);
1390 :                                         [174]
1400 ' *** Steuerung Untermenue ***            [2617]
1410 :                                         [174]
1420 yf=2:yfa=2:PEN#4,15:LOCATE#4,1,yf:PRI     [2244]
NT#4,CHR$(174);
1430 WHILE INKEY$<>"":WEND                     [1786]
1440 IF INKEY(vor) THEN 1460 ELSE yf=yf-1:     [3732]
IF yf<2 THEN yf=2
1450 GOTO 2710                                 [397]
1460 IF INKEY(hin) THEN 1480 ELSE yf=yf+1:     [2755]
IF yf>maxi THEN yf=maxi
1470 GOTO 2710                                 [397]
1480 IF INKEY(feu) THEN 1440 ELSE SOUND 2,     [2289]
250,1,15
1490 IF yf<maxi THEN 1640                      [1654]
1500 CLS#4:PRINT CHR$(22)CHR$(1);              [1510]
1510 set=-1:IF f=13 THEN PEN 3:LOCATE 20,2     [2622]
1:PRINT CHR$(195);:GOTO 1530
1520 DI:ON f GOSUB 4390,4460,4460,4780,478     [2599]
0,4300,4300,4540,4640
1530 PRINT CHR$(22)CHR$(0);                    [1383]
1540 EI:GOSUB 1580:dla=0:nix=-1:GOTO 640       [1972]
1550 :                                         [174]
1560 ' *** Hauptmenue ***                      [954]
1570 :                                         [174]
1580 PAPER#4,7:CLS#4:p=13:xw=10:yw=2:a$="B     [5034]
ISTRO 'LA CUISINE'":GOSUB 5340
1590 p=11:xw=11:yw=4:a$="K KOCHREZEPTE":GO     [5263]
SUB 5340:yw=5
1600 a$="I INVENTUR":GOSUB 5340:yw=6:a$="B     [4051]
 BESTELLUNGEN":GOSUB 5340:RETURN
1610 :                                         [174]
1620 ' *** Nehmen ***                          [972]
1630 :                                         [174]
1640 IF f=4 OR f=5 OR f>9 THEN 1840 ELSE y     [3893]
neu=yf-1
1650 IF itu>2 THEN 1800                        [1262]
1660 IF wt(f,yneu)=0 THEN wt(f,yneu)=-1 EL     [1996]
SE 2710
1670 IF bmk(1)=2 AND (f<2 OR f=8 OR f=9) T     [1620]
HEN 1820
1680 IF f<2 OR f=8 OR f=9 THEN bmk(itu)=2:     [5831]
IF f=1 AND yf=2 THEN ssl=-1:GOTO 1760
1690 IF bmk(itu)=2 THEN IF f=1 AND yf=3 TH     [4300]
EN teller=-1:GOTO 1760 ELSE 1760
1700 IF bmk(1)=1 AND (f>1 AND f<8) THEN 18     [1449]
20
1710 IF f>1 AND f<8 THEN bmk(itu)=1:zt=zt(     [4973]
f,yneu):nm$=in$(f,yneu):nmset=0 ELSE 1760
1720 IF f=yf-1 AND f=3 THEN 1760               [1425]
1730 FOR i=1 TO 10:IF in$(f,yf-1)<>zub$(i,     [3577]
1) THEN NEXT:GOTO 1760
1740 IF num=10 THEN 1820 ELSE num=num+1        [1990]
1750 mk(num)=i:auf=1:df$(num,auf)=zub$(i,1     [4019]
):d1(itu)=num:d2(itu)=auf:nmset=-1:nm$=""
1760 LOCATE#4,2,yf:PRINT#4,STRING$(9,32);:     [3265]
SOUND 2,400,1,15
1770 IF f=9 THEN IF yf=2 THEN wuerz=-1 ELS     [2721]
E messer=-1
1780 IF f=8 THEN IF yf=2 THEN topf=-1 ELSE     [1028]
 pfanne=-1
1790 ivt$(itu)=in$(f,yneu):itu=itu+1:GOTO      [1124]
2710
1800 LOCATE#4,11,6:PRINT#4," ":LOCATE#4,1,     [2929]
1:PRINT#4,CHR$(8)
1810 a$="HAENDE VOLL!":xw=9:yw=2:p=4:GOSUB     [2828]
 5340:GOTO 1440
1820 LOCATE#4,11,6:PRINT#4," ":LOCATE#4,1,     [2929]
1:PRINT#4,CHR$(8)
1830 a$="NICHT MOEGLICH!":xw=9:yw=2:p=5:GO     [3894]
SUB 5340:GOTO 1440
1840 IF f>4 THEN 1880 ELSE yw=yw+1:IF bmk(     [3398]
1)=1 THEN n=1 ELSE n=2
1850 IF d2(n)<1 THEN a$=meld$(2):GOTO 2030     [3955]
 ELSE a$="OK! SENF ENTNOMMEN!"
1860 xw=9:yw=6:p=3:GOSUB 5340:p$="SENF":GO     [5163]
SUB 3060:WHILE INKEY$<>"":WEND
1870 yneu=1:GOTO 1790                          [1113]
1880 IF f>5 THEN 1920 ELSE yw=yw+1:IF bmk(     [2059]
1)=1 THEN n=1 ELSE n=2
```

Listing CUISINE

```
1890 IF d2(n)<1 THEN a$=meld$(2):GOTO 2030 [4229]
 ELSE a$="OK! KETCHUP ENTNOMMEN!"
1900 xw=9:yw=6:p=3:GOSUB 5340:p$="KETCHUP" [4648]
:GOSUB 3060:WHILE INKEY$<>"":WEND
1910 yneu=1:GOTO 1790                      [1113]
1920 IF f>10 THEN 2240 ELSE frei=-1        [1261]
1930 IF bmk(1)=1 THEN a1$=ivt$(1):dw=1 ELS [3094]
E a1$=ivt$(2):dw=2
1940 p=pvoll+1:IF yf=4 THEN spl(5)=d1(dw): [6398]
sp2(5)=d2(dw) ELSE spl(p)=d1(dw):sp2(p)=d2
(dw)
1950 IF yf>2 THEN 2050                     [564]
1960 IF itu=1 OR (itu=2 AND bmk(1)=2) THEN [2391]
 1440
1970 CLS#4:xw=9:yw=4:p=7                   [1681]
1980 IF pvoll=4 THEN a$="ALLE PLATTEN BELE [5138]
GT!":GOSUB 5340:GOTO 2040
1990 IF NOT pfanne THEN a$=meld$(1):GOTO 2 [4518]
030 ELSE a$=a1$:GOSUB 5340
2000 t$=LEFT$(a1$,1):IF t$="N" OR t$="E" T [4062]
HEN a$="WERDEN" ELSE a$="WIRD"
2010 a$=a$+" GEBRATEN!":yw=5:p=9:GOSUB 222 [2147]
0:itu=1:pfanne=0:wt(8,2)=0
2020 FOR i=1 TO 2:ivt$(i)="":bmk(i)=0:NEXT [2149]
:IF NOT nmset THEN GOSUB 3110
2030 GOSUB 5340                            [879]
2040 WHILE INKEY$<>"":WEND:dp=0:GOTO 4040  [1847]
2050 IF yf>3 THEN 2140                     [603]
2060 IF itu=1 OR (itu=2 AND bmk(1)=2) THEN [2391]
 1440
2070 CLS#4:xw=9:yw=4:p=7                   [1681]
2080 IF pvoll=4 THEN a$="ALLE PLATTEN BELE [5138]
GT!":GOSUB 5340:GOTO 2040
2090 IF NOT topf THEN a$=meld$(1):GOTO 203 [3145]
0 ELSE a$=a1$:GOSUB 5340
2100 t$=LEFT$(a1$,1):IF t$="N" OR t$="E" T [4062]
HEN a$="WERDEN" ELSE a$="WIRD"
2110 a$=a$+" GEKOCHT!":yw=5:p=9:GOSUB 2220 [3805]
:itu=1:topf=0:wt(8,1)=0
2120 FOR i=1 TO 2:ivt$(i)="":bmk(i)=0:NEXT [2149]
:IF NOT nmset THEN GOSUB 3110
2130 GOTO 2030                             [365]
2140 IF yf>4 THEN 2740                     [860]
2150 IF itu=1 OR (itu=2 AND bmk(1)=2) THEN [2391]
 1440
2160 t$=LEFT$(a1$,2):IF t$<>"PI" AND t$<>" [4158]
BA" AND t$<>"HA" THEN 1440
2170 CLS#4:xw=9:yw=4:p=7:a$=a1$            [1719]
2180 IF bo$<>"" THEN a$="BACKOFEN IST BELE [4156]
GT!":GOSUB 5340:GOTO 2040
2190 IF bmk(1)=1 THEN ivt$(1)="":bmk(1)=0: [5208]
itu=1 ELSE ivt$(2)="":bmk(2)=0:itu=2
2200 GOSUB 5340:a$="WIRD GEBACKEN!":yw=5:p [4076]
=9:IF NOT nmset THEN GOSUB 3110
2210 bo$=LEFT$(a1$,8)+"-BA":start(5)=mi:da [4559]
uer(5)=zt:GOTO 2030
2220 pvoll=pvoll+1:plat$(pvoll)=LEFT$(a1$, [4826]
8)+"-"+LEFT$(in$(f,yf-1),2)
2230 start(pvoll)=mi:dauer(pvoll)=zt:pgk(p [4145]
voll)=gk:RETURN
2240 IF bmk(1)=1 THEN a$=ivt$(1):n=1 ELSE  [2596]
a$=ivt$(2):n=2
2250 IF f>11 THEN 2320                     [1042]
2260 IF itu=1 OR (itu=2 AND bmk(1)=2) THEN [2391]
 1440
2270 t$=LEFT$(a$,2):IF t$="FE" OR t$="SU"  [3419]
THEN 1440
2280 CLS#4:xw=9:yw=4:p=7:GOSUB 5340        [2086]
2290 IF t$="EI" OR t$="NU" THEN a$="SIND"  [4003]
ELSE a$="IST"
2300 a$=a$+" GEWASCHEN!":yw=5:p=9:p$=" -WA [2492]
SCHEN":GOSUB 3060
2310 GOTO 2030                             [365]
2320 IF f>12 THEN 2640                     [714]
2330 IF bmk(1)=1 THEN a1$=ivt$(1):dw=1:n=2 [2283]
 ELSE a1$=ivt$(2):dw=2:n=1
2340 spl=d1(dw):sp2=d2(dw)                 [810]
2350 IF yf<5 THEN IF itu=1 OR (itu=2 AND b [2290]
mk(1)=2) THEN 1440
2360 IF yf>2 THEN 2430                     [648]
2370 IF NOT messer THEN CLS#4:a$=meld$(1): [3258]
yw=7:xw=10:GOSUB 5340:GOTO 3410
2380 messer=0:CLS#4:xw=9:yw=4:p=7:GOSUB 53 [1871]
40
2390 t$=LEFT$(a$,2):IF t$="EI" OR t$="NU"  [3132]
THEN a$="SIND" ELSE a$="IST"
```

Listing CUISINE

```
2400 ivt$(n)="":wt(9,2)=0:n=spl            [2176]
2410 a$=a$+" GESCHNITTEN!":yw=5:p=9:p$=" - [5047]
SCHNEIDEN":IF t$="BR" THEN IF NOT nmset TH
EN GOSUB 3110
2420 GOTO 2480                             [395]
2430 IF yf>3 THEN 2500                     [595]
2440 IF NOT wuerz THEN CLS#4:a$=meld$(1):y [3520]
w=7:xw=10:GOSUB 5340:GOTO 3410
2450 wuerz=0:wt(9,1)=0:CLS#4:xw=9:yw=4:p=7 [3078]
:GOSUB 5340
2460 t$=LEFT$(a$,2):IF t$="EI" OR t$="NU"  [3132]
THEN a$="SIND" ELSE a$="IST"
2470 ivt$(n)="":n=spl:a$=a$+" GEWUERZT!":y [6633]
w=5:p=9:p$=" -WUERZEN":IF t$="HA" THEN d2(
n)=d2(n)+1
2480 GOSUB 3060:GOSUB 3080                 [1867]
2490 GOTO 2030                             [365]
2500 IF yf>4 THEN 2680                     [904]
2510 IF df$(1,1)="" THEN v$=df$(2,1) ELSE  [2683]
v$=df$(1,1)
2520 IF v$="REIS" THEN supp=1 ELSE supp=0  [2332]
2530 IF tsh THEN IF bmk(1)=1 THEN dw=1:GOT [4702]
O 2570 ELSE dw=2:GOTO 2570
2540 IF supp=1 THEN IF ssl=0 THEN 2560 ELS [3448]
E 2570
2550 IF teller THEN 2570                   [1235]
2560 CLS#4:a$=meld$(2):yw=7:xw=10:GOSUB 53 [4169]
40:GOTO 3410
2570 IF df$(0,1)="" THEN cz=1              [887]
2580 CLS#4:xw=9:yw=4:p=7:GOSUB 5340:n=spl: [3602]
tsh=-1:FOR i=1 TO sp2+1
2590 IF df$(n,i)<>"" THEN df$(0,dfw+cz)=df [2438]
$(n,i):cz=cz+1
2600 df$(n,i)="":NEXT:IF dfw>11 THEN 1830  [1740]
2610 t$=LEFT$(a$,2):dp=-1:nmset=0          [1961]
2620 IF t$="EI" OR t$="NU" THEN a$="SIND"  [4003]
ELSE a$="IST"
2630 a$=a$+" PLATZIERT!":yw=5:p=9:GOSUB 53 [2999]
40:'GOSUB 3050
2640 itu=1:ivt$(1)="":bmk(1)=0:ivt$(2)="": [4904]
bmk(1)=0:IF dp THEN 2040 ELSE SOUND 7,2000
,50,,1,,4
2650 topf=0:pfanne=0:teller=0:ssl=0        [1667]
2660 IF bef THEN FOR i=0 TO 4:FOR j=1 TO 1 [3485]
1:df$(i,j)="":NEXT j:d1(i)=0:d2(i)=0:NEXT:
bef=0:nmset=0:cz=1:num=0
2670 FOR i=1 TO 13:FOR j=1 TO 4:wt(i,j)=0: [3091]
NEXT j,i:IF dla THEN 1540 ELSE 1500
2680 CLS#4:xw=9:yw=4:p=6:a$="ABLAGE IST GE [4003]
SAEUBERT!":GOSUB 5340
2690 teller=0:FOR i=1 TO 13:FOR j=1 TO 4:w [3791]
t(i,j)=0:NEXT j,i
2700 FOR c=0 TO 10:df$(0,c)="":d1(c)=0:d2( [3960]
c)=0:NEXT:dfw=0:GOTO 2040
2710 SOUND 2,80,1,4:PEN#4,15:LOCATE#4,1,yf [1812]
a
2720 PRINT#4," ";:LOCATE#4,1,yf:PRINT#4,CH [2804]
R$(174);
2730 yfa=yf:FOR is=1 TO 100:NEXT:GOTO 1440 [2294]
2740 CLS#4:xw=9:yw=1:p=12:FOR i=1 TO 4:yw= [4900]
yw+1:a$="P"+MID$(STR$(i),2,2)+":"
2750 IF plat$(i)="" THEN a$=a$+"*FREI*":GO [3305]
TO 2770 ELSE a$=a$+plat$(i):frei=0
2760 GOSUB 3010                            [967]
2770 GOSUB 5340:NEXT                       [1221]
2780 i=5:a$="B5:":IF bo$="" THEN a$=a$+"*F [3828]
REI*":GOTO 2800 ELSE a$=a$+bo$:frei=0
2790 GOSUB 3010                            [967]
2800 yw=yw+1:GOSUB 5340:IF frei THEN 3410  [2241]
2810 a$="N-NEHMEN ODER R-RAUS":yw=yw+1:p=1 [3617]
:GOSUB 5340
2820 a$=UPPER$(INKEY$):IF a$<>"N" AND a$<> [2783]
"R" THEN 2820
2830 IF a$="R" THEN 1540 ELSE xw=10:yw=7:p [1730]
=4
2840 LOCATE#4,1,6:PRINT#4,STRING$(10,32);  [1596]
2850 IF a=5 THEN pt=spl(5) ELSE pt=spl(pvo [2084]
ll)
2860 a$="WAEHLE 1-5":xw=11:yw=7:GOSUB 5340 [2517]
2870 b$=INKEY$:IF b$<"1" OR b$>"5" THEN 28 [3890]
70 ELSE a=VAL(b$)
2880 LOCATE#4,1,6:PRINT#4,STRING$(10,32);: [3744]
IF a=5 AND bo$>"" THEN 2900
2890 IF plat$(a)="" THEN 1540 ELSE xw=10:y [1794]
w=7:p=4
2900 IF itu=3 THEN a$="HAENDE VOLL!":GOSUB [1589]
 5340:GOTO 3410
```

Listing CUISINE

```
2910 IF bmk(1)=1 THEN a$="NICHT MOEGLICH!" [2268]
:GOSUB 5340:GOTO 3410
2920 IF a<5 THEN a$="PLATTE "+b$ ELSE a$=" [1340]
BACKOFEN"
2930 a$=a$+" IST GEWAEHLT!":xw=9:GOSUB 534 [2499]
0
2940 bmk(itu)=1:ivt$(itu)=plat$(a):d1(itu) [5210]
=sp1(a):d2(itu)=sp2(a):plat$(a)=""
2950 IF a<5 THEN FOR i=1 TO 4:IF plat$(i)= [2762]
"" THEN NEXT:pvoll=pvoll-1
2960 IF a=5 THEN ivt$(itu)=bo$:bo$=""      [1245]
2970 r$=RIGHT$(ivt$(itu),2):IF r$="BR" THE [4938]
N p$=" -BRATEN:0":GOTO 2990
2980 IF r$="BA" THEN p$=" -BACKEN:0" ELSE  [2217]
p$=" -KOCHEN:0"
2990 IF ok(a) THEN p$=p$+MID$(STR$(dauer(a [2729]
)),2,2)
3000 n=itu:GOSUB 3060:itu=itu+1:GOTO 3410  [2069]
3010 ok(i)=0:zw=mi-start(i):zr=start(i)+da [3480]
uer(i)-mi
3020 IF zw<dauer(i) THEN a$=a$+" *0"+MID$( [3339]
STR$(zr),2,2)+"*":RETURN
3030 IF zw>dauer(i) THEN a$=a$+" *VB*" ELS [2882]
E a$=a$+" *OK*":ok(i)=-1
3040 RETURN                                [555]
3050 d1(n)=sp1(a):d2(n)=sp2(a)             [1663]
3060 num=d1(n):auf=d2(n):auf=auf+1:IF auf> [4180]
11 THEN PRINT CHR$(7);:GOTO 2040
3070 df$(num,auf)=p$:d2(n)=auf:RETURN      [2824]
3080 IF itu=3 THEN IF bmk(1)=1 THEN 3100 E [2547]
LSE ivt$(1)=ivt$(2):ivt$(2)=""
3090 bmk(1)=bmk(2):d1(1)=d1(2):d2(1)=d2(2) [3881]
:bmk(2)=0:d1(2)=0:d2(2)=0
3100 itu=2:RETURN                          [1008]
3110 df$(d1(dw),d2(dw))=nm$:nmset=-1:d2(dw [4515]
)=d2(dw)+1:RETURN
3120 :                                     [174]
3130 ' *** Kochrezepte ***                 [589]
3140 :                                     [174]
3150 PAPER#4,3:CLS#4:IF bst=0 THEN 3180    [1844]
3160 a$="R - REZEPT FUER":xw=10:yw=2:p=9:G [2306]
OSUB 5340
3170 a$=grt$:xw=12:yw=3:GOSUB 5340         [1517]
3180 xw=10:yw=5:a$="A - ALLE REZEPTE":GOSU [1613]
B 5340
3190 a$="BITTE WAEHLEN":xw=11:yw=7:p=8:GOS [4197]
UB 5340
3200 a1$=UPPER$(INKEY$):IF a1$<>"R" AND a1 [2367]
$<>"A" THEN 3200
3210 IF a1$="R" THEN 3560 ELSE FOR i=1 TO  [4131]
4:LOCATE#4,11,6:PRINT#4,"  ":NEXT
3220 FOR i=1 TO 4:LOCATE#4,1,1:PRINT#4,CHR [2696]
$(8):NEXT:p=2
3230 a$="B - BILDSCHIRM":xw=11:yw=3:GOSUB  [3180]
5340:xw=14:yw=4:a$="ODER"
3240 GOSUB 5340:xw=11:yw=5:a$="D - DRUCKER [3089]
":GOSUB 5340
3250 a2$=UPPER$(INKEY$):IF a2$<>"B" AND a2 [3058]
$<>"D" THEN 3250
3260 IF a2$="B" THEN 3310                  [1303]
3270 IF a2$="D" THEN 3450 ELSE 1540        [1215]
3280 :                                     [174]
3290 ' *** Liste auf Bildschirm ***        [2192]
3300 :                                     [174]
3310 CLS#4:WINDOW#5,9,19,3,7:PAPER#5,3:FOR [3907]
 i=1 TO agr:a$="REZEPT: "+rp$(i)
3320 a$="REZEPT: "+rp$(i):xw=9:yw=2:p=14:G [2577]
OSUB 5340
3330 FOR j=1 TO zbs(i):IF j<>5 AND j<>9 TH [3006]
EN 3350
3340 a$="*TASTE*":xw=11:yw=7:p=13:GOSUB 53 [3148]
40:WHILE INKEY$="":WEND:yw=2:CLS#5
3350 xw=10:yw=yw+1:p=12:a$=zub$(i,j):GOSUB [4013]
 5340:NEXT j:yw=7:xw=11:p=1
3360 IF such THEN yw=7:such=0:GOTO 3400    [2086]
3370 a$="* W - WEITER *":GOSUB 5340        [1029]
3380 a$=UPPER$(INKEY$):IF a$="" THEN 3380  [1924]
3390 IF a$="W" THEN CLS#4:NEXT i:yw=4 ELSE [2323]
 1540
3400 a$="*ENDE REZEPTLISTE*":xw=10:p=14:GO [3821]
SUB 5340
3410 WHILE INKEY$<>"":WEND:WHILE INKEY$="" [4173]
:WEND:GOTO 1540
3420 :                                     [174]
3430 ' *** Liste auf Drucker ***           [1380]
3440 :                                     [174]
3450 PRINT#8,CHR$(27);"W";CHR$(1);' Breit  [2067]
ein
```

Listing CUISINE

```
3460 PRINT#8,TAB(9);"REZEPTLISTE FUER 'LA  [3847]
CUISINE'"
3470 PRINT#8,CHR$(27);"W";CHR$(0);' Breit  [2809]
aus
3480 PRINT#8:FOR i=1 TO agr                [881]
3490 PRINT#8,CHR$(27);"4" 'Kursiv ein      [1571]
3500 PRINT#8,TAB(30);"REZEPT: ";rp$(i);:PR [3589]
INT#8,CHR$(27);"5" 'Kursiv aus
3510 PRINT#8:FOR j=1 TO zbs(i):PRINT#8,TAB [2009]
(34);zub$(i,j):NEXT j
3520 PRINT#8:NEXT i:GOTO 1540              [1645]
3530 :                                     [174]
3540 ' *** Aktuelles Rezept***             [1701]
3550 :                                     [174]
3560 CLS#4:WINDOW#5,9,19,3,7:PAPER#5,3:suc [4603]
h=-1:i=akt(bst):GOTO 3320
3570 :                                     [174]
3580 ' *** Inventur ***                    [934]
3590 :                                     [174]
3600 PAPER#4,9:CLS#4:a$="INVENTAR:-":xw=9: [4853]
yw=2:p=15:GOSUB 5340
3610 lz=0:p=12:FOR i=1 TO itu:yw=yw+1:IF i [4958]
vt$(i)>"" THEN lz=lz+1
3620 a$=ivt$(i):GOSUB 5340:NEXT:IF lz>0 TH [1325]
EN 3640
3630 yw=4:a$="* NICHTS *":GOSUB 5340       [1810]
3640 a$="*ENDE INVENTUR*":xw=10:yw=7:p=13: [1539]
GOSUB 5340
3650 WHILE INKEY$<>"":WEND:WHILE INKEY$="" [4173]
:WEND:GOTO 1540
3660 :                                     [174]
3670 ' *** Liste Bestellungen ***          [2123]
3680 :                                     [174]
3690 IF bst=0 THEN PRINT CHR$(7);:GOTO 640 [2681]
3700 PAPER#4,4:CLS#4:xw=9:yw=1:p=13:FOR i= [3984]
1 TO bst:yw=yw+1:a$=STR$(i)
3710 a$=a$+" - "+bst$(i):GOSUB 5340:NEXT   [1667]
3720 yw=yw+1:p=10:a$="F-FERTIG ODER R-RAUS [4288]
":GOSUB 5340
3730 a$=UPPER$(INKEY$):IF a$<>"F" AND a$<> [1991]
"R" THEN 3730
3740 IF a$="R" THEN 1540 ELSE LOCATE#4,1,y [3759]
w-1:PRINT#4,STRING$(10,32);
3750 a$="WAEHLE 1 -"+STR$(bst):xw=10:GOSUB [4149]
 5340:p=7:PAPER#4,1
3760 a$=INKEY$:IF a$<"1" OR a$>MID$(STR$(b [2667]
st),2,2) THEN 3760
3770 :                                     [174]
3780 ' *** Bestellung fertig ***           [2629]
3790 :                                     [174]
3800 gr=VAL(a$):CLS#4:a$="* "+bst$(gr)+" * [3385]
":xw=10:yw=2:p=2:GOSUB 5340
3810 bef=-1:schr=0:IF bmk(1)=1 THEN n=d1(1 [4253]
) ELSE n=d1(2)
3820 FOR j=1 TO zbs(akt(gr)):IF zub$(akt(g [3876]
r),j)=df$(0,j) THEN schr=schr+1
3830 NEXT:IF schr<>zbs(akt(gr)) THEN 3900  [2793]
3840 a$="DAS GERICHT IST O.K.!":xw=9:yw=3: [3141]
p=8:GOSUB 5340
3850 a$="GAST IST ZUFRIEDEN UND":yw=4:p=10 [2578]
:GOSUB 5340
3860 a$="WILL DAS BISTRO WEIT-":yw=5:GOSUB [2925]
 5340
3870 a$="EREMPFEHLEN!!":yw=6:GOSUB 5340    [2886]
3880 FOR i=1 TO 8:FOR j=400 TO 100 STEP -5 [4040]
*i:SOUND 1,j,1,15:NEXT j,i
3890 gzuf=gzuf+1:GOTO 3980                 [1504]
3900 IF schr>zbs(akt(gr))/2 AND schr<zbs(a [3350]
kt(gr)) THEN 3930
3910 a$="GERICHT IST VOELLIGER":xw=9:yw=3: [3826]
p=7:GOSUB 5340
3920 a$="SCHROTT !!!!":yw=4:GOSUB 5340:GOT [3893]
O 3950
3930 a$="GERICHT NICHT VOLL-":xw=9:yw=3:p= [3079]
13:GOSUB 5340
3940 a$="STAENDIG !!":yw=4:GOSUB 5340      [1310]
3950 a$="GAST IST UNZUFRIEDEN":yw=5:p=14:G [3016]
OSUB 5340
3960 a$="UND VERLAESST BISTRO.":yw=6:GOSUB [2187]
 5340
3970 FOR i=0 TO 1000 STEP 10:SOUND 2,500+i [4569]
,4,15:NEXT:gver=gver+1
3980 LOCATE#2,platz(gr),3:PRINT#2," ";:LOC [3171]
ATE 7,7:PRINT CHR$(22)CHR$(0);:PEN 10
3990 FOR i=1 TO dfw:df$(0,i)="":NEXT:dfw=0 [2868]
4000 PRINT CHR$(143);:xw=7:yw=7:a$=MID$(ST [5047]
R$(bst-1),2,2):p=8:snd=0:GOSUB 5340
```

Listing CUISINE

```
4010 IF bst>1 AND gnz=0 THEN FOR i=gr TO b [7029]
st:bst$(i)=bst$(i+1):platz(i)=platz(i+1):a
kt(i)=akt(i+1):NEXT
4020 bst=bst-1:snd=-1:dla=-1:IF bst=0 THEN [2593]
 CLS#3:grt$=""
4030 IF gnz=1 THEN gnz=0:tim=0             [1482]
4040 nix=-1:a$="*TASTE*"::xw=11:yw=7:p=13: [4138]
GOSUB 5340:WHILE INKEY$="":WEND
4050 IF dla THEN 2640 ELSE 1540            [1198]
4060 IF bmk(1)=1 THEN dw=1 ELSE dw=2       [1616]
4070 :                                     [174]
4080 ' *** Screen-Aufbau ***               [1144]
4090 :                                     [174]
4100 MODE 0:WINDOW#1,1,20,1,9:PAPER#1,9:PE [2396]
N#1,10:CLS#1
4110 PRINT#1,CHR$(22)CHR$(1):LOCATE#1,1,1: [2880]
FOR i=1 TO 9
4120 PRINT#1,STRING$(20,166);:NEXT         [1182]
4130 WINDOW#1,1,20,10,23:PAPER#1,13:CLS#1: [2536]
PEN#1,10
4140 FOR i=1 TO 14:PRINT#1,STRING$(20,165) [2862]
;:NEXT
4150 LOCATE 1,24:PEN 11:PAPER 0:PRINT STRI [3379]
NG$(20,207);
4160 PRINT STRING$(20,216);:PAPER 0        [1917]
4170 WINDOW#2,2,6,2,5:PAPER#2,5:PEN#2,4:CL [2187]
S#2
4180 PRINT#2,CHR$(167);" ";CHR$(167);" ";C [2504]
HR$(167);
4190 PRINT#2:PRINT#2:PEN#2,8:PRINT#2,STRIN [2984]
G$(5,143);
4200 PLOT 34,334,14:DRAW 190,334:PEN 6:LOC [3253]
ATE 6,12:PRINT CHR$(143);
4210 PAPER 0:WINDOW#3,4,17,9,9:PAPER#3,10: [2281]
CLS#3
4220 xw=2:yw=7:LOCATE xw,yw:PEN 10:PRINT S [4249]
TRING$(6,143);
4230 a$="BESTELL.:":p=7:snd=0:GOSUB 5340   [2099]
4240 WINDOW#4,9,19,2,7:GOSUB 1580:snd=-1   [2498]
4250 GOSUB 4300:GOSUB 4390:GOSUB 4460:GOSU [3711]
B 4540
4260 GOSUB 4640:GOSUB 4800:PRINT CHR$(22)C [2295]
HR$(0);
4270 FOR i=0 TO 2 STEP 2:PLOT 1,254-i,9:DR [2929]
AW 640,254-i:NEXT
4280 tp=-1:GOSUB 5110:TAGOFF:tg=0:GOTO 487 [1904]
0
4290 ' *** Kuehlschrank ***                [1695]
4300 LOCATE 1,1:PRINT CHR$(22)CHR$(1);:LOC [1876]
ATE 5,17:PEN 15
4310 PRINT CHR$(169)CHR$(143)CHR$(170):FOR [2607]
 i=18 TO 24
4320 LOCATE 5,i:PRINT STRING$(3,143);:PEN  [2553]
6
4330 IF i=24 THEN PRINT CHR$(173):GOTO 435 [2550]
0
4340 IF i=18 THEN PRINT CHR$(172) ELSE PRI [3773]
NT CHR$(171)
4350 PEN 15:NEXT:PLOT 213,70,2:DRAW 213,90 [3501]
:PLOT 216,76,4:DRAW 216,88
4360 PLOT 213,105,2:DRAW 213,120:PLOT 216, [3480]
110,4:DRAW 216,118
4370 PLOT 130,96,3:DRAW 220,96:RETURN      [1297]
4380 ' *** Wandschrank ***                 [1398]
4390 FOR i=15 TO 18:LOCATE 1,i:PEN 2:PRINT [2881]
 STRING$(3,143);
4400 PEN 6:IF A=18 THEN PRINT CHR$(173);:G [2310]
OTO 4420
4410 IF i>15 THEN PRINT CHR$(171); ELSE PR [3461]
INT CHR$(172);
4420 NEXT:PLOT 46,112,3:DRAW 46,174        [1960]
4430 PLOT 32,145,15:PLOT 58,145:PLOT 32,14 [2286]
3,7:PLOT 58,143
4440 PLOT 1,174,6:DRAW 92,174:PLOT 1,112:D [3411]
RAW 92,112:RETURN
4450 ' *** unterer Schrank ***             [1827]
4460 FOR i=20 TO 24:LOCATE 1,i:PEN 7:PRINT [4476]
 STRING$(3,143);:PEN 6
4470 IF i<>20 AND i<>24 THEN PRINT CHR$(17 [3236]
1);:GOTO 4490
4480 IF i=20 THEN PRINT CHR$(172); ELSE IF [3580]
 i=24 THEN PRINT CHR$(173);
4490 NEXT:a(1)=94:a(2)=80:a(3)=78:a(4)=16: [2188]
FOR i=1 TO 4
4500 PLOT 1,a(i),11:DRAW 92,a(i):NEXT      [2761]
4510 PLOT 46,16:DRAW 46,78:PLOT 46,86,9:PL [2162]
OT 46,88,15:PLOT 30,58,6
```

Listing CUISINE

```
4520 PLOT 60,58:PLOT 30,60,15:PLOT 60,60:R [1580]
ETURN
4530 ' *** Spuele ***                      [622]
4540 FOR i=20 TO 24:LOCATE 9,i:PEN 11:PRIN [4339]
T STRING$(4,143);
4550 PEN 6:IF i=24 THEN PRINT CHR$(173);:G [3566]
OTO 4570
4560 IF i=20 THEN PRINT CHR$(172); ELSE PR [2837]
INT CHR$(171);
4570 NEXT:PEN 3:LOCATE 11,19:PRINT CHR$(17 [3744]
5)
4580 a(1)=94:a(2)=48:a(3)=16:FOR i=1 TO 3: [3037]
PLOT 256,a(i),14
4590 DRAW 382,a(i):NEXT                    [241]
4600 LOCATE 11,15:PEN 3:PRINT CHR$(177):LO [4402]
CATE 11,16:PEN 8
4610 PRINT CHR$(176):PEN 15:LOCATE 10,15:P [3422]
RINT CHR$(177)
4620 PEN 8:LOCATE 10,16:PRINT CHR$(176):RE [2811]
TURN
4630 ' *** Herd ***                        [636]
4640 LOCATE 15,14:PEN 13:PRINT STRING$(3,1 [3394]
43);:PEN 6:PRINT CHR$(172)
4650 PEN 8:LOCATE 14,15:PRINT CHR$(214);ST [3695]
RING$(3,143);CHR$(215)
4660 IF set THEN set=0:RETURN              [1461]
4670 a$=STRING$(5,143):PEN 13:LOCATE 14,20 [4074]
:PRINT a$:LOCATE 14,24:PRINT a$
4680 PEN 6:LOCATE 15,20:PRINT CHR$(181)CHR [2132]
$(182)CHR$(181)
4690 FOR i=21 TO 23:LOCATE 14,i:PEN 13:PRI [2637]
NT CHR$(133);:PEN 9
4700 PRINT CHR$(8);CHR$(138);              [1368]
4710 PRINT STRING$(3,143);CHR$(133);:PEN 1 [3099]
3:PRINT CHR$(8);CHR$(138);
4720 PEN 6:PRINT CHR$(171);:NEXT           [1829]
4730 LOCATE 19,20:PRINT CHR$(172):LOCATE 1 [4183]
9,24:PRINT CHR$(173)
4740 PEN 5:LOCATE 17,12:PRINT CHR$(150);ST [2294]
RING$(3,154);
4750 LOCATE 17,13:PRINT CHR$(149)          [1558]
4760 FOR i=0 TO 2:PLOT 432,50+i*10,2:DRAW  [2629]
556,50+i*10:NEXT
4770 PLOT 432,32,1:DRAW 432,78:DRAW 560,78 [3435]
:DRAW 560,32,7:DRAW 436,32
4780 RETURN                                [555]
4790 ' *** Muelleimer ***                  [1598]
4800 LOCATE 20,21:PEN 3:PRINT CHR$(195);:F [2868]
OR i=22 TO 24:LOCATE 20,i
4810 PEN 7:PRINT CHR$(143);:NEXT:FOR x=608 [4103]
 TO 640 STEP 8
4820 PLOT x,16,9:DRAW x,60:NEXT:PLOT 608,1 [3655]
6:DRAW 640,16
4830 PLOT 608,62:DRAW 640,62:RETURN        [2627]
4840 :                                     [174]
4850 ' *** Uhr & Screen ***                [1373]
4860 :                                     [174]
4870 ab=REMAIN(1):SOUND 64,40,1,15:EVERY 9 [2494]
00,2 GOSUB 5100
4880 OUT &BC00,1:FOR i=1 TO 40:OUT &BD00,i [4502]
:FOR j=1 TO 20:NEXT j,i:x=10:y=15
4890 GOSUB 4930:GOTO 640                   [1683]
4900 :                                     [174]
4910 ' *** Gast in Bistro ***              [1399]
4920 :                                     [174]
4930 DI:IF bst=5 THEN 5060                 [1031]
4940 pl=INT(RND*5+1):FOR h=1 TO bst:IF pla [3769]
tz(h)=pl OR pl>5 THEN 4940 ELSE NEXT
4950 IF tg=-1 THEN sp=1:tg=0:TAGOFF        [1992]
4960 IF cw=1 THEN cw=0:so=1:LOCATE 1,1:PRI [2951]
NT CHR$(23)CHR$(cw);
4970 bst=bst+1:gins=gins+1:LOCATE 7,7:PEN  [3415]
10:PRINT CHR$(143);:CLS#3
4980 platz(bst)=pl:LOCATE#2,platz(bst),3:P [3245]
EN#2,1:PRINT#2,CHR$(168);
4990 xq1=xw1:yq1=yw1:xq=xw:yq=yw:pq=p:aq$= [4003]
a$:tj=rz
5000 xw=7:yw=7:a$=MID$(STR$(bst),2,2):p=8: [3867]
snd=0:GOSUB 5340:snd=-1
5010 IF cw=0 AND so=1 THEN cw=1:so=0:LOCAT [4064]
E 1,1:PRINT CHR$(23)CHR$(cw);
5020 IF tg=0 AND sp=1 THEN tg=-1:sp=0:TAG  [2030]
5030 zf=INT(RND*10+1):IF lev=1 AND zf>8 TH [4877]
EN 5030 ELSE grt$=rp$(zf):akt(bst)=zf:bst$
(bst)=grt$
5040 w5=xw1:w6=yw1:p7=p:z$="* "+grt$+" *": [5329]
k1=5:l1=9:fp1=10:GOSUB 5460
```

Listing CUISINE

```
5050 xw=xq:yw=yq:p=pq:xwl=xql:ywl=yql:a$=a  [6504]
q$:rz=tj:IF txt THEN PLOT -2,-2,p:MOVE xwl
+rz*16,ywl
5060 EVERY 7700-lev*(RND*1000+1000),1 GOSU  [3544]
B 4930:EI:RETURN
5070 :                                      [174]
5080 ' *** Uhr ***                          [888]
5090 :                                      [174]
5100 DI:hm=0:st$="":mi$="":tp=0:mi=mi+1:IF  [3354]
 mi>=60 THEN mi=0:st=st+1:tp=-1
5110 IF tg=-1 THEN tg=0:hm=-1:TAGOFF        [1724]
5120 IF cw=1 THEN w3(2)=-1:cw=0:LOCATE 1,1  [3967]
:PRINT CHR$(23)CHR$(cw);
5130 IF tg=0 THEN w3(1)=-1:tg=-1:TAG        [2269]
5140 IF st=24 THEN st=0:mi=0:tp=-1          [1789]
5150 IF mi<10 THEN mi$="0":mm=1 ELSE mi$="  [2712]
":mm=2
5160 IF st<10 THEN st$=" 0":ms=3 ELSE st$=  [3065]
" ":ms=2
5170 mi$=mi$+MID$(STR$(mi),2,mm):st$=st$+M  [4570]
ID$(STR$(st),2,ms)
5180 PLOT -2,-2,13:IF NOT tp THEN 5200      [2277]
5190 FOR z1=1 TO 3:MOVE 128+z1*16,222:PRIN  [3279]
T MID$(st$,z1,1);:NEXT
5200 FOR z2=1 TO 2:MOVE 176+z2*16,222:PRIN  [4014]
T MID$(mi$,z2,1);:NEXT
5210 PLOT 192,214,3:PLOT 192,218:PLOT -2,-  [2979]
2,p:MOVE xwl+rz*16,ywl
5220 IF tg=-1 AND w3(1)=-1 THEN w3(1)=0:tg  [2088]
=0:TAGOFF
5230 IF cw=0 AND w3(2)=-1 THEN w3(2)=0:cw=  [3807]
1:LOCATE 1,1:PRINT CHR$(23)CHR$(cw);
5240 IF tg=0 AND hm=-1 THEN tg=-1:TAG       [1684]
5250 EI:w3(1)=0:w3(2)=0:IF st=el AND mi=e2  [2809]
 THEN 5580
5260 IF gnz=0 AND bst=5 THEN gnz=1:tim=mi   [1667]
5270 IF gnz=1 AND tim+4=mi THEN CLS#4:a$="  [5371]
BAD LUCK MY FRIEND!" ELSE RETURN
5280 xw=9:yw=2:p=6:GOSUB 5340:a$="DU WARST  [4968]
 ZU LANGSAM.":yw=4:GOSUB 5340
5290 a$="DIE GAESTE HABEN DIE":yw=5:GOSUB   [4128]
5340:a$="NASE VOLL UND SIND WEG":yw=6:GOSU
B 5340
5300 LOCATE#2,1,3:PRINT#2,STRING$(5,32):FO  [8403]
R i=1 TO 5:bst$(i)="":platz(i)=0:akt(i)=0:
NEXT:gver=gver+4:bst=0:gr=1:bst=1:platz(1)
=1:GOTO 3970
5310 :                                      [174]
5320 ' *** Text ausgeben ***                [1134]
5330 :                                      [174]
5340 txt=-1:hb=0:IF tg=-1 THEN tg=0:hb=-1:  [2462]
TAGOFF
5350 IF cw=0 THEN w2(2)=-1:cw=1:LOCATE 1,1  [2594]
:PRINT CHR$(23)CHR$(cw);
5360 IF tg=0 THEN w2(1)=-1:tg=-1:TAG        [1873]
5370 ywl=400-yw*16+14:xwl=xw*32-48:FOR rz=  [3212]
1 TO LEN(a$)
5380 PLOT -2,-2,p:MOVE xwl+rz*16,ywl:PRINT  [2991]
 MID$(a$,rz,1);
5390 IF snd THEN SOUND 2,100,1,4            [1290]
5400 NEXT:IF tg=-1 AND w2(1)=-1 THEN tg=0:  [1946]
w2(1)=0:TAGOFF
5410 IF cw=1 AND w2(2)=-1 THEN w2(2)=0:cw=  [4650]
0:LOCATE 1,1:PRINT CHR$(23)CHR$(cw);
5420 IF tg=0 AND hb=-1 THEN tg=-1:TAG       [2331]
5430 w2(1)=0:w2(2)=0:txt=0:RETURN           [2136]
5440 LOCATE x,y:PEN 1:PRINT CHR$(178):LOCA  [2262]
TE x,y+1:PEN 13:PRINT CHR$(179)
5450 LOCATE x,y+2:PEN 1:PRINT CHR$(180):RE  [2210]
TURN
5460 IF tg THEN tg=0:hx=-1:TAGOFF           [1467]
5470 IF cw=0 THEN w3=-1:cw=1:LOCATE 1,1:PR  [2721]
INT CHR$(23)CHR$(cw);
5480 IF tg=0 THEN w4=-1:tg=-1:TAG           [1943]
5490 ml=400-l1*16+14:nl=kl*32-48:FOR mn=1   [3068]
TO LEN(z$)
5500 PLOT -2,-2,fpl:MOVE nl+mn*16,ml:PRINT  [2615]
 MID$(z$,mn,1);
5510 SOUND 1,50,1,5:NEXT:IF tg AND w4 THEN  [3482]
 tg=0:TAGOFF
5520 IF cw=1 AND w3 THEN cw=0:LOCATE 1,1:P  [3541]
RINT CHR$(23)CHR$(cw);
5530 IF tg=0 AND hx THEN tg=-1:TAG          [1256]
5540 w4=0:w3=0:hx=0:RETURN                  [925]
5550 :                                      [174]
5560 ' *** Auswertung ***                   [1386]
5570 :                                      [174]
```

Listing CUISINE

```
5580 tg=0:TAGOFF:cw=0:PRINT CHR$(23);CHR$(  [3102]
0);:a=REMAIN(1):PEN 1
5590 b=REMAIN(2):LOCATE 4,9:PRINT CHR$(22)  [5863]
CHR$(0);" *SPIELENDE* "
5600 FOR i=1 TO 3:SOUND 2,200,70,,1:NEXT:F  [3828]
OR i=1 TO 4000:NEXT
5610 FOR i=1 TO 25:FOR j=1 TO i:LOCATE 1,1  [2728]
:PRINT CHR$(8):NEXT:FOR j=1 TO i
5620 LOCATE 20,25:PRINT CHR$(9):NEXT j,i:y  [4630]
=8:PLOT 1,1,5:DRAW 1,399
5630 DRAW 639,399:DRAW 639,1:DRAW 1,1:FOR   [2553]
x=2 TO 6 STEP 2:GOSUB 5440
5640 y=y-3:NEXT:a$="A U S W E R T U N G":x  [4150]
w=9:yw=3:p=1:GOSUB 5340
5650 a$=STRING$(23,45):xw=8:yw=4:GOSUB 534  [2851]
0
5660 a$="BESTELLUNGEN INSGESAMT:"+STR$(gin  [5797]
s):p=10:xw=6:yw=9:GOSUB 5340
5670 a$="ZUFRIEDENE GAESTE :"+STR$(gzuf):p  [3412]
=11:yw=11:GOSUB 5340
5680 a$="VERAERGERTE GAESTE:"+STR$(gver):p  [3584]
=12:yw=13:GOSUB 5340
5690 a$="SCHWIERIGKEITSGRAD:"+STR$(lev):p=  [3283]
13:yw=16:GOSUB 5340
5700 a$="GESAMTSCORE        :"+STR$(500*lev  [3265]
*gzuf):p=15:yw=18:GOSUB 5340
5710 a$="*** "+urt$(gzuf/2)+" ***":p=1:yw=  [2333]
20:GOSUB 5340
5720 a$="NOCHEINMAL ?":p=8:yw=23:xw=9:GOSU  [3221]
B 5340
5730 a$=UPPER$(INKEY$):IF a$<>"J" AND a$<>  [1595]
"N" THEN 5730
5740 IF a$="J" THEN RUN ELSE MODE 2:END     [2269]
5750 ' Farben & Urteile                     [1467]
5760 DATA 26,12,6,2,13,0,3,9,7,23,11,27,17  [6876]
,19,21,MIES,SCHWACH,NICHT UEBEL,STARK,SUPE
R,SPITZE,CHEFKOCH
5770 ' Kueche                               [143]
5780 DATA WANDSCHRANK,15,18,1,3,2,SUPPENSC  [2659]
HUESSEL,TELLER
5790 DATA MEHRZWECKFACH1,20,20,1,3,2,FERTI  [3635]
GSOSSE,SUPPENEXTRAKT
5800 DATA BODENSCHRANK,21,24,1,3,3,NUDELN,  [3653]
REIS,BROETCHEN
5810 DATA SPENDER1,15,16,10,10,1,SENF       [2155]
5820 DATA SPENDER2,15,16,11,11,1,KETCHUP    [1709]
5830 DATA KUEHLSCHRANK,20,24,5,7,4,HAEHNCH  [2423]
EN,EIER,SPECK,WUERSTCHEN
5840 DATA GEFRIERFACH,18,19,5,7,4,PIZZA,BA  [3710]
GUETTE,HAMBURGER,FISCHFILET
5850 DATA BEHAELTERDEPOT,23,24,9,12,2,TOPF  [2853]
,PFANNE
5860 DATA MEHRZWECKFACH2,14,14,15,17,2,GEW  [4138]
UERZE,KUECHENMESSER
5870 DATA HERD,20,20,15,17,4,BRATEN,KOCHEN  [3962]
,BACKEN,UEBERSICHT
5880 DATA SPUELE,19,19,11,11,1,WASCHEN      [2553]
5890 DATA ABLAGE,19,19,9,10,4,SCHNEIDEN,WU  [2899]
ERZEN,PLATZIEREN,ABLAGE NEU
5900 DATA MUELLEIMER,21,24,20,20,1,INVENTA  [1687]
R HINEIN
5910 ' Rezepte                              [346]
5920 DATA HOT DOG,5,BROETCHEN," -SCHNEIDEN  [4345]
",WUERSTCHEN," -BRATEN:03",SENF
5930 DATA EIER MIT SPECK,5,SPECK," -SCHNEI  [4124]
DEN"," -BRATEN:01",EIER," -BRATEN:02"
5940 DATA HLB. HAEHNCHEN,5,HAEHNCHEN," -WA  [3524]
SCHEN"," -SCHNEIDEN"," -WUERZEN"," -BACKEN
:04"
5950 DATA PIZZA,2,PIZZA," -BACKEN:04"       [1385]
5960 DATA SPAGHETTI,5,NUDELN," -KOCHEN:04"  [5839]
,FERTIGSOSSE," -KOCHEN:02"," -WUERZEN"
5970 DATA BIG HAM,6,HAMBURGER," -BRATEN:03  [4532]
"," -WUERZEN",BROETCHEN," -SCHNEIDEN",KETC
HUP
5980 DATA REISSUPPE,5,REIS," -WASCHEN"," -  [5656]
KOCHEN:04",SUPPENEXTRAKT," -KOCHEN:01"
5990 DATA BAGUETTE,2,BAGUETTE," -BACKEN:04  [3070]
"
6000 DATA FISCH MIT REIS,11,FISCHFILET," -  [9897]
WASCHEN"," -SCHNEIDEN"," -BRATEN:04"," -WU
ERZEN",REIS," -WASCHEN"," -KOCHEN:04",FERT
IGSOSSE," -KOCHEN:02"," -WUERZEN"
6010 DATA WUERST.GULASCH,9,WUERSTCHEN," -B  [8378]
RATEN:04"," -SCHNEIDEN",REIS," -WASCHEN","
 -KOCHEN:04",FERTIGSOSSE," -KOCHEN:02"," -
WUERZEN"
```

Listing CUISINE

Der Zeitwürfel

Zeitspiele mit dem CPC

Sinn des Spiels ist es, alle Zeiger der beiden 9er Blöcke auf 12 Uhr zu stellen. Dabei hat man eine Vielzahl von Möglichkeiten, die Uhren zu verdrehen. Zum Spielen sind insgesamt zwölf Tasten zu gebrauchen. Mit den Cursortasten links beziehungsweise rechts lassen sich die Uhren vor- oder zurückstellen. Die Nummern eins bis vier im rechten Zahlenblock beziehungsweise die Tasten Q,W,E,R dienen zur Auswahl des zu drehenden Pins, die Zahlen eins bis vier am oberen Tastenfeld werden zur Auswahl einer Drehmöglichkeit benutzt. Um bei dem aktuellen Stand der Zeiger nächstes Mal weiterspielen zu können, muß *CTRL+S* gedrückt werden. Den Spielstand des letzten Spiels kann man durch *CTRL+L* wieder laden.
Und nun ran ans Werk!

(Volker Nordhoff/jb)

für 464-664-6128

```
10 '******************************          [1223]
20 '**     RUBIK'S CLOCK ON CPC    **        [2529]
30 '**       Volker Nordhoff       **        [1680]
40 '**          CPC 6128           **        [668]
50 '******************************          [1223]
60 INK 0,0:INK 1,15:INK 2,26:INK 3,8:BORDE [1639]
R 0
70 SYMBOL AFTER 32:tr=HIMEM+1              [2815]
80 SYMBOL 91,255,65,65,65,65,65,127,8      [2710]
90 SYMBOL 93,255,127,127,127,127,127,127,8 [2429]
100 SYMBOL 169,76,210,68,72,94             [1728]
110 aq$="1234qwer"+CHR$(19)+CHR$(12)+CHR$( [2458]
242)+CHR$(243)
120 DIM f(15,4,9,2),po(9,2,2),ek(4,2),az1( [3627]
4),az2(4),zei(9,2)
130 DEF FNte=az1(1)+az1(2)+az1(3)+az1(4)   [2217]
140 GOSUB 430                              [974]
150 FOR t=1 TO 9:FOR k=1 TO 2:zei(t,k)=INT [4617]
(RND*11)+1:NEXT:NEXT
160 zei(1,2)=zei(1,1):zei(3,2)=zei(3,1):ze [3353]
i(7,2)=zei(7,1):zei(9,2)=zei(9,1):MODE 1
170 LOCATE 11,1:t$="RUBIK'S CLOCK ON CPC   [2127]
180 PEN 2:FOR c=1 TO 20:dr=tr+(ASC(MID$(t$ [14125]
,c,1))-32)*8:FOR w=0 TO 7:r(w)=PEEK(dr+w):
NEXT:SYMBOL 250,0,r(0),r(0),r(1),r(1),r(2)
,r(2),r(3):SYMBOL 251,r(3),r(4),r(4),r(5),
r(5),r(6),r(6),r(7):PRINT CHR$(250)CHR$(10
)CHR$(8)CHR$(251)CHR$(11);:NEXT
190 DEG:FOR i=1 TO 9:FOR k=1 TO 2:FOR t=30 [1665]
 TO 360 STEP 30
200 TAG:PLOT po(i,1,k)+33*SIN(t),po(i,2,k) [5876]
+33*COS(t),3:IF t<360 THEN PRINT CHR$(144)
;ELSE PRINT CHR$(169);
210 TAGOFF                                 [1066]
220 NEXT:PLOT po(i,1,k)+6,po(i,2,k)-6,2:DR [4663]
AWR 16*SIN(zei(i,k)*30),16*COS(zei(i,k)*30
):NEXT:NEXT
230 FOR t=26 TO 330 STEP 304:PLOT t,50,1:D [9209]
RAWR 0,278:DRAWR 280,0:DRAWR 0,-278:DRAWR
-280,0:PLOT t-4,46:DRAWR 0,286:DRAWR 288,0
:DRAWR 0,-286:DRAWR -288,0:NEXT
240 FOR t=88 TO 250 STEP 96                [1060]
250 PLOT t+32,50:DRAWR 0,278:PLOT t+336,50 [6891]
:DRAWR 0,278:PLOT 28,t+54:DRAWR 276,0:PLOT
 332,t+54:DRAWR 276,0:NEXT
260 az2(1)=1:az2(2)=2:az2(3)=4:az2(4)=8    [3134]
270 PEN 1:FOR t=1 TO 4:az1(t)=az2(t):LOCAT [8060]
E ek(t,1),ek(t,2)-1:PRINT MID$(STR$(t),2,1
)CHR$(8)CHR$(10)"]":LOCATE ek(t,1)+19,ek(t
,2)-1:PRINT MID$(STR$(t),2,1)CHR$(8)CHR$(1
0)"[";:NEXT
280 anz=1:t=1:GOSUB 680:KEY 129,"q":KEY 13 [5034]
0,"w":KEY 131,"e":KEY 132,"r"
290 a$=INKEY$:FOR t=1 TO 12:IF MID$(aq$,t, [2560]
1)<>a$ THEN NEXT:GOTO 290
300 IF a$=CHR$(12)THEN OPENIN"rub.dat":FOR [5744]
 t=1 TO 9:FOR k=1 TO 2:INPUT#9,zei(t,k):NE
XT:NEXT:CLOSEIN:GOTO 160
310 IF a$=CHR$(19)THEN OPENOUT"rub.dat":FO [6048]
R t=1 TO 9:FOR k=1 TO 2:PRINT#9,zei(t,k):N
EXT:NEXT:CLOSEOUT:GOTO 290
320 FOR t=1 TO 4:IF VAL(a$)=t THEN IF az1( [7522]
t)=0 THEN az1(t)=az2(t):GOSUB 690 ELSE az1
(t)=0:GOSUB 690 ELSE NEXT
330 FOR t=1 TO 4:IF a$=MID$(aq$,t+4,1)THEN [5338]
 GOSUB 680 ELSE NEXT
```

Listing Zeitwürfel

```
340 IF a$=CHR$(242)THEN dd=-1 ELSE IF a$=C [2743]
HR$(243)THEN dd=1 ELSE 290
350 za=0:as=FNte:FOR k%=1 TO 2:FOR a%=1 TO [2051]
 9
360 IF f(as,anz,a%,k%)=0 THEN 390 ELSE i=f [3032]
(as,anz,a%,k%)
370 PLOT po(i,1,k%)+6,po(i,2,k%)-6,0:DRAWR [10906]
 16*SIN(zei(i,k%)*30),16*COS(zei(i,k%)*30)
:zei(i,k%)=zei(i,k%)+dd:IF zei(i,k%)=13 TH
EN zei(i,k%)=1 ELSE IF zei(i,k%)=0 THEN ze
i(i,k%)=12
380 PLOT po(i,1,k%)+6,po(i,2,k%)-6,2:DRAWR [6026]
 16*SIN(zei(i,k%)*30),16*COS(zei(i,k%)*30)
390 za=za+zei(i,k%):IF za=216 THEN 410     [1988]
400 NEXT:NEXT:GOTO 290                     [1363]
410 TAG:PLOT 160,354,3:PRINT"Du bist der C [2733]
HAMPION";:TAGOFF
420 PEN 2:LOCATE 16,25:PRINT"SPIELENDE":LO [4342]
CATE 1,24:END
430 FOR t=0 TO 15:FOR k=1 TO 4:FOR i=1 TO  [2570]
9
440 READ f(t,k,i,1):f(15-t,k,i,2)=f(t,k,i, [1729]
1)
450 NEXT:NEXT:NEXT                         [963]
460 DATA 1,3,7,9,0,0,0,0,0,1,3,7,9,0,0,0,0 [3122]
,0,1,3,7,9,0,0,0,0,0,1,3,7,9,0,0,0,0,0
470 DATA 1,2,4,5,0,0,0,0,0,3,7,9,0,0,0,0,0 [3642]
,0,3,7,9,0,0,0,0,0,3,7,9,0,0,0,0,0,0
480 DATA 1,7,9,0,0,0,0,0,0,2,3,5,6,0,0,0,0 [3375]
,0,1,7,9,0,0,0,0,0,1,7,9,0,0,0,0,0,0
490 DATA 1,2,3,4,5,6,0,0,0,1,2,3,4,5,6,0,0 [3051]
,0,7,9,0,0,0,0,0,0,7,9,0,0,0,0,0,0,0
500 DATA 1,3,9,0,0,0,0,0,0,1,3,9,0,0,0,0,0 [3150]
,0,4,5,7,8,0,0,0,0,1,3,9,0,0,0,0,0,0
510 DATA 1,2,4,5,7,8,0,0,0,3,9,0,0,0,0,0,0 [3176]
,0,1,2,4,5,7,8,0,0,0,3,9,0,0,0,0,0,0
520 DATA 1,9,0,0,0,0,0,0,0,2,3,4,5,6,7,8,0 [3476]
,0,2,3,4,5,6,7,8,0,0,1,9,0,0,0,0,0,0
530 DATA 1,2,3,4,5,6,7,8,0,1,2,3,4,5,6,7,8 [3388]
,0,1,2,3,4,5,6,7,8,0,9,0,0,0,0,0,0,0
540 DATA 1,3,7,0,0,0,0,0,0,1,3,7,0,0,0,0,0 [3689]
,0,1,3,7,0,0,0,0,0,5,6,8,9,0,0,0,0,0
550 DATA 1,2,4,5,6,8,9,0,0,3,7,0,0,0,0,0,0 [3045]
,0,3,7,0,0,0,0,0,0,1,2,4,5,6,8,9,0,0
560 DATA 1,7,0,0,0,0,0,0,0,2,3,5,6,8,9,0,0 [3759]
,0,1,7,0,0,0,0,0,0,2,3,5,6,8,9,0,0,0
570 DATA 1,2,3,4,5,6,8,9,0,1,2,3,4,5,6,8,9 [2778]
,0,7,0,0,0,0,0,0,0,1,2,3,4,5,6,8,9,0
580 DATA 1,3,0,0,0,0,0,0,0,1,3,0,0,0,0,0,0 [3804]
,0,4,5,6,7,8,9,0,0,4,5,6,7,8,9,0,0,0
590 DATA 1,2,4,5,6,7,8,9,0,3,0,0,0,0,0,0,0 [3746]
,0,1,2,4,5,6,7,8,9,0,1,2,4,5,6,7,8,9,0
600 DATA 1,0,0,0,0,0,0,0,0,2,3,4,5,6,7,8,9 [3524]
,0,2,3,4,5,6,7,8,9,0,2,3,4,5,6,7,8,9,0
610 DATA 1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8 [3298]
,9,1,2,3,4,5,6,7,8,9,1,2,3,4,5,6,7,8,9
620 FOR t=1 TO 9:READ a,b                  [1552]
630 po(t,1,1)=a:po(t,2,1)=b:po(t,1,2)=a+30 [2335]
4:po(t,2,2)=b
640 NEXT                                   [350]
650 DATA 64,290,160,290,256,290,64,194,160 [2915]
,194,256,194,64,98,160,98,256,98
660 FOR t=1 TO 4:READ a,b:ek(t,1)=a:ek(t,2 [3609]
)=b:NEXT:RETURN
670 DATA 8,11,14,11,8,17,14,17             [338]
680 FOR ds=0 TO 19 STEP 19:LOCATE ek(anz,1 [10852]
)+ds,ek(anz,2)-1:PRINT CHR$(48+anz):LOCATE
 ek(t,1)+ds,ek(t,2)-1:PRINT CHR$(24)CHR$(4
8+t)CHR$(24):NEXT:anz=t:RETURN
690 PEN 1:IF az1(t)>0 THEN LOCATE ek(t,1), [8058]
ek(t,2):PRINT"]":LOCATE ek(t,1)+19,ek(t,2)
:PRINT"["ELSE LOCATE ek(t,1),ek(t,2):PRINT
"[":LOCATE ek(t,1)+19,ek(t,2):PRINT"]"
700 RETURN                                 [555]
```

Listing Zeitwürfel

Mein CPC ist voll!!!

Strategiespiel mit Überraschungen

Eimerkette ist ein Spiel für alle CPC-Rechner, welches nach einem einfachen (Überlauf-)Prinzip arbeitet. Dadurch daß, wenn ein Eimer voll ist, dessen überschüssiger Inhalt in andere Eimer verteilt werden muß, ist dieses Spiel nur auf einem Computer realisierbar. (Wer will schon seine Wohnung unter Wasser setzen?)

Das Spielfeld (sechs mal sechs Felder groß) wird wie ein Schachfeld an den horizontalen und vertikalen Seiten mit a-f und 1-6 beschriftet. Der eigentliche Sinn des Spieles liegt darin, seinen Gegenspieler, der auch vom Computer simuliert werden kann, vollkommen vom Spielfeld zu verdrängen. Das heißt jeder Spieler setzt abwechselnd seine "Eimer" auf die günstigsten Felder und füllt diese dadurch mit Wasser. Je voller ein Eimer wird, desto größer ist auch die Möglichkeit des Überlaufens, und da beginnt eigentlich auch schon der Reiz dieses Spiels. Da jeder Spieler seine "Eimer" auf das Spielfeld setzt und nach und nach diese dann mit Wasser füllt, wodurch es dann zum Überlauf kommt, verteilen sich die "Eimer" auf ihre benachbarten Felder. Durch diese Verteilung vermehrt sich der Besitzanteil eines jeden Spielers. Wird der Eimer des Gegners gefüllt, geht dieser automatisch in den eigenen Besitz über.

Wenn es nun einem Spieler gelingt, alle "Eimer" des Gegners in seinen Besitz zu bringen, ist für ihn das Spiel gewonnen und es kann entweder eine neue Partie stattfinden oder das Spiel verlassen werden.

Auch nach mehrmaligem Spielen verspricht das Programm, durch immer wieder neue Züge des Gegners, interessant zu sein und wird Sie sicherlich für einige Zeit am Computer fesseln.

In das Programm ist auch ein Demonstrationsmodus eingebaut, der beim genaueren Hinsehen so manchen wertvollen Tip geben kann.

Da das Programm voll in Basic geschrieben ist, sollte es auch für den Anfänger leicht zu verstehen sein.

(Stefan Achilles/jb)

für 464-664-6128

```
10 MODE 0                                      [507]
20 LOCATE 6,12:PRINT"Eimerkette                [1881]
30 BORDER 26,2:INK 1,26,2:INK 0,2,26:SPEED     [12941]
 INK 1,2:FOR i=1 TO 2000:SPEED INK 1,7:NEX
T i:FOR i=1 TO 1000:SPEED INK 1,12:NEXT i:
SPEED INK 1,20:FOR i=1 TO 1000:NEXT i:SPEE
D INK 1,10:FOR i=1 TO 1000:NEXT i:SPEED IN
K 1,50:FOR i=1 TO 1000:NEXT i:CALL &BC02
40 CALL &BC02                                  [353]
50 DATA 2,3,3,3,3,2                            [654]
60 REM                                         [272]
70 REM Initialisierung                         [2397]
80 REM                                         [272]
90 MODE 1                                      [506]
100 WINDOW #1,1,40,20,24:WINDOW #2,1,40,25     [2705]
,25
110 CLS:RANDOMIZE TIME*2:LOCATE 16,9:PRINT     [5014]
 "Eimerkette"
120 LOCATE 16,15:PRINT CHR$(164);" 1989 by     [6263]
":PEN 3:LOCATE 16,17:PRINT"DMV Verlag":PEN
 1
130 DIM w(6,6,3),cn$(1),v(2,1)                 [1414]
140 FOR i=1 TO 6:READ w(1,i,0):w(6,i,0)=w(     [3254]
1,i,0):FOR j=2 TO 5:w(j,i,0)=w(1,i,0)+1
150 NEXT j:NEXT i                              [825]
160 DEF FN rn(x)=INT(RND*x+1)                  [124]
180 REM                                        [272]
190 REM Spielvorbereitung                      [1469]
200 REM                                        [272]
210 LOCATE 11,11:PRINT"Willkommen zum Spie     [2939]
l"
220 LOCATE 8,24:PRINT CHR$(24): INPUT"Brau     [4608]
chst du Instructionen ";x$:PRINT CHR$(24)
230 IF LEFT$(x$,1)="J" OR LEFT$(X$,1)="j"      [2064]
THEN GOSUB 1040
240 CLS:PRINT "Zahl der Menschlichen Spiel     [4806]
er: [0,1,2]"
250 x$=INKEY$:IF x$="" OR x$<"0" OR x$>"2"     [3646]
 THEN 250 ELSE g=VAL(x$)
260 IF g<2 THEN cn$(1)="CPC>464<":IF g<2 T     [3696]
HEN cn$(0)="Amstrad":GOTO 310
270 FOR i=1 TO g                               [431]
280 LOCATE 2,4*i:PRINT "Name des Spielers"     [4107]
;i;:INPUT x$
290 cn$(i-1)=LEFT$(x$,10)                      [1217]
300 NEXT i:m=0                                 [321]
310 CLS:GOSUB 690:IF RND<0.5 THEN k=1 ELSE     [3336]
 k=2
```

Listing Eimerkette

```
320 REM                                        [272]
330 REM Hauptprogramm                          [1411]
340 REM                                        [272]
350 CLS #1                                     [373]
360 m=m+1:s=1-(k=1)                            [793]
370 IF k>10 THEN k=10 ELSE IF k<1 THEN k=1     [2721]
380 PEN k+1:PEN #1,k+1:LOCATE #1,4,1:PRINT     [3310]
 #1,cn$(k-1);"'s Zug:":IF k=2 AND g<2 THEN
 1140
390 IF k=1 AND g=0 THEN 1140                   [810]
400 LOCATE #1,4,3:PRINT #1,"Zeile?"            [1977]
410 x$=INKEY$:IF x$="" OR x$<"1" OR x$>"6"     [3435]
 THEN 410
420 y=VAL(x$):LOCATE #1,11,3:PRINT #1,x$       [1939]
430 LOCATE #1,14,3:PRINT #1,"Spalte?"          [1804]
440 x$=INKEY$:IF x$="" OR x$<"a" OR x$>"f"     [3968]
 THEN 440
450 x=ASC(x$)-96:LOCATE #1,22,3:PRINT #1,x     [3794]
$
460 IF w(x,y,2)=k OR w(x,y,1)=0 THEN 500       [2158]
470 SOUND 1,110:LOCATE #1,1,4:PRINT CHR$(2     [7948]
4):PRINT #1,"verbotene POSITION - noch mal
...":PRINT CHR$(24)
480 FOR i=1 TO 1000:NEXT i                     [674]
490 CLS #1:GOTO 380                            [551]
500 w(x,y,1)=w(x,y,1)+1:w(x,y,2)=k:GOSUB 7     [4410]
70:IF w(x,y,1)=1 THEN v(k,1)=v(k,1)+1
510 z=1:IF w(x,y,1)=w(x,y,0) THEN GOSUB 86     [1633]
0
520 IF z<>1 THEN 580                           [801]
530 IF k=1 THEN k=2 ELSE k=1                   [648]
540 GOTO 350                                   [470]
550 REM                                        [272]
560 REM Spielende                              [383]
570 REM                                        [272]
580 PRINT #1,REMAIN (0):CLS #1:LOCATE #1,3     [5450]
,2:PRINT #1,cn$(k-1);" hat gewonnen !!!":P
RINT #1:PRINT #1,"Noch mal?"
590 v(k,0)=v(k,0)+1:PEN 2:LOCATE 32,10:PRI     [7031]
NT STR$(v(1,0)):PEN 3:LOCATE 32,12:PRINT S
TR$(v(2,0)):GOSUB 770
600 x$=INKEY$:IF x$="" THEN 600                [2676]
610 IF x$="N" OR x$="n" THEN GOTO 1630         [1494]
620 CLS #1:FOR i=1 TO 6:FOR j=1 TO 6:w(j,i     [3424]
,1)=0:w(j,i,2)=0:NEXT j:NEXT i:v(1,1)=0:v(
2,1)=0
630 m=0:mi=0:se=0:s1=0:s2=0:m1=0:m2=0          [1282]
640 FOR i=1 TO 6:FOR j=1 TO 6:LOCATE 2*j+6     [3010]
,2*i+3:PRINT " ":NEXT j:NEXT i
650 CLS #2:EVERY 50,0 GOSUB 1560:GOTO 350      [2041]
660 REM                                        [272]
670 REM sub - Spielbrett-Aufbau                [1175]
```

Listing Eimerkette

```
680 REM                                     [272]
690 PEN 1:LOCATE 7,1:PRINT "*EimER-keTte*"  [9192]
:FOR i=0 TO 12 STEP 2:MOVE 104,343-i*16:DR
AW 296,343-i*16:NEXT i:FOR i=0 TO 12 STEP
2:MOVE 104+i*16,343:DRAW 104+i*16,152:NEXT
 i
700 MOVE 76,375:DRAW 324,375:DRAW 324,123:  [7770]
DRAW 76,123:DRAW 76,375:FOR i=0 TO 5:FOR j
=0 TO 1:LOCATE 6+j*14,5+i*2:PRINT CHR$(i+4
9):LOCATE 8+i*2,3+j*14:PRINT CHR$(i+65):NE
XT j:NEXT i
710 PEN 1:LOCATE 23,8:PRINT "Spielstand":P  [6659]
EN 2:LOCATE 23,10:PRINT cn$(0):PEN 3:LOCAT
E 23,12:PRINT cn$(1)
720 PEN 2:LOCATE 32,10:PRINT STR$(v(0,0)):  [6350]
PEN 3:LOCATE 32,12:PRINT STR$(v(1,0))
730 EVERY 50,0 GOSUB 1560:RETURN            [2661]
740 REM                                     [272]
750 REM sub - setzen                        [619]
760 REM                                     [272]
770 FOR i=1 TO 6:FOR j=1 TO 6               [1924]
780 IF w(j,i,1)=0 THEN LOCATE 2*j+6,2*i+3:  [3029]
PRINT " ":GOTO 810
790 IF w(j,i,2)=2 THEN PEN 3 ELSE PEN 2     [2103]
800 LOCATE 2*j+6,2*i+3:PRINT MID$(STR$(w(j  [2845]
,i,1)),2,1)
810 NEXT j:NEXT i                           [825]
820 RETURN                                  [555]
830 REM                                     [272]
840 REM sub - ueberlauf                     [609]
850 REM                                     [272]
860 FOR y=1 TO 6:FOR x=1 TO 6               [1449]
870 IF w(x,y,1)>=w(x,y,0) THEN 890          [1267]
880 NEXT x:NEXT y:RETURN                    [1503]
890 PEN k+1                                 [486]
900 FOR i=136 TO 138:LOCATE 2*x+6,2*y+3:PR  [1678]
INT CHR$(i)
910 FOR j=1 TO 200:NEXT j:NEXT i            [1886]
920 IF y>1 THEN q=y-1:w(x,q,1)=w(x,q,1)+1:  [3146]
u=x:GOSUB 1500
930 IF x>1 THEN u=x-1:w(u,y,1)=w(u,y,1)+1:  [3863]
q=y:GOSUB 1500
940 IF y<6 THEN q=y+1:w(x,q,1)=w(x,q,1)+1:  [3417]
u=x:GOSUB 1500
950 IF x<6 THEN u=x+1:w(u,y,1)=w(u,y,1)+1:  [1703]
q=y:GOSUB 1500
960 w(x,y,1)=w(x,y,1)-w(x,y,0):w(x,y,2)=k   [3797]
970 IF w(x,y,1)=0 THEN w(x,y,2)=0:v(k,1)=v  [2859]
(k,1)-1
980 GOSUB 770                               [931]
990 z=1:IF (v(1,1)+v(2,1)>1) AND (v(1,1)=0  [4270]
 OR v(2,1)=0) THEN z=0:RETURN
1000 GOTO 860                               [536]
1010 REM                                    [272]
1020 REM Instruktionen                      [1539]
1030 REM                                    [272]
1040 CLS                                    [91]
1050 PRINT TAB(13);"SPieLanLEItunG":PRINT:  [3262]
PRINT
1060 PRINT "  Die beiden SPIELER legen ABW  [15057]
ECHSELND":PRINT TAB (10);"Steine AUF das F
ELD.":PRINT:PRINT " Aus Gemeinheit darf ma
n diese aber nur":PRINT "   auf LEERE oder
EIGENE Felder setzen."
1070 PRINT:PRINT TAB(7);"Da jedes Feld nur  [17883]
 BEGRENZT":PRINT TAB(6);"aufnahmefaehig is
t, laeuft es":PRINT "  ploetzlich UEBER un
d VERTEILT dabei":PRINT "  seinen INHALT gl
eichmaessig auf seine":PRINT TAB(11);"NACH
BARN (WUERG)."
1080 PRINT:PRINT " GEGNERISCHE Felder wech  [13565]
seln dabei die":PRINT "  FARBE und somit d
ie ZUGEHOERIGKEIT.":PRINT:PRINT "Das ZIEL
ist es, den Gegner RESTLOS vom":PRINT TAB(
11);          "FELD zu draengen."
1090 PRINT:PRINT TAB(14);CHR$(24)"Alles kl  [6161]
ar?":PRINT CHR$(24):PRINT "  Dann drueck'
doch 'mal 'ne TASTE !!!"
1100 IF INKEY$="" THEN 1100 ELSE RETURN     [2116]
1110 REM                                    [272]
1120 REM Compu-Gambler                      [1304]
1130 REM                                    [272]
1140 q=0:u=0:FOR i=1 TO 6:FOR j=1 TO 6      [2004]
1150 w(j,i,3)=0:IF w(j,i,2)=k AND w(j,i,1)  [5089]
=w(j,i,0)-1 THEN w(j,i,3)=1:q=q+1
```

Listing Eimerkette

```
1160 IF w(j,i,2)<>s THEN u=u+1              [1802]
1170 NEXT j:NEXT i:IF q=0 THEN 1320         [1386]
1180 FOR i=1 TO 6:FOR j=1 TO 6:IF w(j,i,3)  [3125]
=0 THEN i=0:j=0:GOTO 1280
1190 ox=j:FOR px=i-1 TO 1 STEP -1:GOSUB 14  [5851]
50:IF w(j,px,1)<>w(j,px,0)-1 THEN px=1
1200 NEXT px:IF i>5 THEN px=1:GOSUB 1230    [1995]
1210 FOR px=i+1 TO 6:GOSUB 1450:IF w(j,px,  [2279]
1)<>w(j,px,0)-1 THEN px=6
1220 NEXT px                                [448]
1230 px=i:FOR ox=j TO 1 STEP -1:GOSUB 1450  [2748]
:IF w(ox,i,1)<>w(ox,i,0)-1 THEN ox=1
1240 NEXT ox:IF j>5 THEN 1270               [725]
1250 FOR ox=j+1 TO 6:GOSUB 1450:IF w(ox,i,  [3626]
1)<>w(ox,i,0)-1 THEN ox=6
1260 NEXT ox                                [426]
1270 NEXT j:NEXT i                          [825]
1280 m=0:FOR px=1 TO 6:FOR ox=1 TO 6        [1916]
1290 IF w(ox,px,3)>m THEN m=w(ox,px,3):i=p  [4153]
x:j=ox
1300 NEXT ox:NEXT px:IF q<>1 OR ((i=1 OR i  [4518]
=6) AND (j=1 OR j=6) AND rn(2)=1) THEN 141
0
1310 IF q=1 AND w(j,i,3)>50 THEN 1410       [1332]
1320 b3=0:IF rn(3)=1 OR u>33 THEN 1350      [1483]
1330 FOR px=1 TO 6:FOR ox=1 TO 6:IF w(ox,p  [4712]
x,2)=k THEN i=px:j=ox:px=6:ox=6
1340 NEXT ox:NEXT px:GOTO 1360              [2097]
1350 i=FN rn(6):j=FN rn(6):IF w(j,i,2)=s T  [2304]
HEN 1350
1360 r=0:FOR px=i-1 TO i-(i<6): IF w(j,px,  [2235]
2)<>s THEN 1380
1370 IF w(j,px,1)=w(j,px,0)-1 THEN r=1      [1802]
1380 NEXT px:FOR ox=j-1 TO j-(j<6):IF w(ox  [4167]
,i,2)<>s THEN 1400
1390 IF w(ox,i,1)=w(ox,i,0)-1 THEN r=1      [1663]
1400 NEXT ox:IF r=1 THEN b3=b3+1:IF b3<u T  [1763]
HEN 1350
1410 LOCATE #1,20,1:PRINT #1,i,CHR$(j+96):  [3717]
x=j:y=i:GOTO 500
1420 REM                                    [272]
1430 REM sub - Punkte                       [403]
1440 REM                                    [272]
1450 w(j,i,3)=w(j,i,3)+1:                   [3978]
  IF w(ox,px,2)=s THEN w(j,i,3)=w(j,i,3)+
50
1460 RETURN                                 [555]
1470 REM                                    [272]
1480 REM sub - Erhoehung Nachbarfeld        [1470]
1490 REM                                    [272]
1500 IF w(u,q,2)=s THEN v(s,1)=v(s,1)-1     [3082]
1510 IF w(u,q,2)<>k THEN v(k,1)=v(k,1)+1    [2324]
1520 w(u,q,2)=k:RETURN                      [1399]
1530 REM                                    [272]
1540 REM Uhr                                [412]
1550 REM                                    [272]
1560 LOCATE #2,18,1:se=se+1:IF se=60 THEN   [1952]
se=0:mi=mi+1
1570 PEN #2,1:PRINT #2,mi;":";se;           [1929]
1580 LOCATE #2,2+(k-1)*30,1:PEN #2,k+1      [2399]
1590 IF k=2 THEN 1610 ELSE s1=s1+1:IF s1=6  [2787]
0 THEN s1=0:m1=m1+1
1600 PRINT #2,m1;":";s1;:RETURN             [1575]
1610 s2=s2+1:IF s2=50 THEN s2=0:m2=m2+1     [2303]
1620 PRINT #2,m2;":";s2;:RETURN             [1478]
1630 REM                                    [272]
1640 REM Ende                               [410]
1650 REM                                    [272]
1660 MODE 1:BORDER 26:INK 0,26:INK 1,0:INK  [3613]
 2,6:INK 3,24
1670 FOR a=-160 TO 160 STEP 8               [1110]
1680 PLOT a*2+320,400,1:DRAW a+320,250:DRA  [3353]
WR 0,-100:DRAW a*2+320,0:NEXT
1690 FOR a=-80 TO 80 STEP 16                [2217]
1700 PLOT 0,a*2+200:DRAW 160,a/2+200:DRAWR  [3453]
 320,0:DRAW 640,a*2+200:NEXT
1710 x=20:FOR a=1 TO 53:y=180-x:PLOT y,y:D  [5738]
RAWR 640-2*y,0:DRAWR 0,400-2*y:DRAWR 2*y-6
40,0:DRAWR 0,2*y-400:x=x*1.1:NEXT
1720 PEN 2                                  [548]
1730 LOCATE 13,13:PRINT"Auf Wiedersehen!"   [1931]
1740 LOCATE 13,24:PRINT CHR$(24)"Bitte ein  [4916]
e Taste":PRINT CHR$(24)
1750 CALL &BB18:CALL 0                      [1280]
```

Listing Eimerkette

Betriebssystem-CALLs für den BASIC-Programmierer

Der CPC verfügt über ein außerordentlich gut organisiertes Betriebssystem: Nahezu alle wichtigen Funktionen können als Unterprogramme mit normierten Einsprungadressen aufgerufen werden. Im folgenden finden Sie eine komplette Aufstellung der CALLs, die sich direkt von BASIC aus benutzen lassen.

Viele Betriebssystem-Routinen erwarten bestimmte Zahlenwerte als Vorgabe in den Z80-Registern oder geben Werte in den Registern zurück. Leider ist dem BASIC-Programmierer ein Zugriff auf die Prozessor-Innereien versagt; ein Aufruf führt hier zu unkontrollierbaren Zufallsergebnissen. Abhilfe kann in diesen Fällen nur ein kleines Maschinenprogramm schaffen, das die Übergabe der Werte organisiert. Allerdings bleiben noch genug Systemroutinen ohne solche Einsprungbedingungen übrig, die sich direkt von BASIC aus aufrufen lassen. Weiterhin ermöglicht es ein kleiner Trick, zumindest das A-Register (Akkumulator) zu beeinflussen: Es enthält nämlich bei einem BASIC-CALL die Anzahl der übergebenen Parameter. Dazu ein Beispiel: Durch

CALL adresse,1,1,1,1

wird das A-Register mit dem Wert 4 geladen. Die Parameter selbst sind hierbei vollkommen belanglos; anstelle der 1 könnte man also jeden beliebigen Wert einsetzen. Die folgende Aufstellung beschränkt sich auf wirklich nutzbringende System-aufrufe. Verzichtet wurde auf CALLs, deren Wirkung sich auch durch einen simplen BASIC-Befehl erreichen läßt. Falls der Parameter-Trick zur Anwendung kommt, wird der CALL-Befehl mit dem Zusatz 'par' versehen. Der Name hinter den CALLs ist die offizielle von Amstrad in der Firmware-Dokumentation verwendete Bezeichnung des Einsprungs.

CALL &0000: RESET ENTRY

setzt den Rechner wie bei CTRL SHIFT ESC komplett zurück. Diese Routine wird manchmal unbeabsichtigt mit fatalen Folgen aufgerufen, wenn ein Programmierer sein Werk gut lesbar gestalten will und deshalb die CALL-Adressen in Variablen ablegt, also zum Beispiel mit CursorOn = &BB81, um im Programm CALL CursorOn schreiben zu können. Ein kleiner Tippfehler wie CALL CursorOm erzeugt dann einen Reset, da die 'unbekannte' Variable mit 0 vorbelegt ist!

CALL &BB00: KM INITIALIZE

initialisiert die Tastaturverwaltung des CPC komplett. Alle Funktionstastenbelegungen, Tastendefinitionen und Wiederholgeschwindigkeiten werden auf die Standardwerte zurückgesetzt und CAPS LOCK ausgeschaltet. Weiterhin werden Unterbrechungen mit ESC gesperrt (siehe auch KM DISARM BREAK).

CALL &BB06: KM WAIT CHAR
CALL &BB18: KM WAIT KEY

Beide Routinen warten auf eine Tastatureingabe und ersetzen die BASIC-Schleife WHILE INKEY$="":WEND. Der erste CALL reagiert dabei auf ein **Zeichen**, also bei Funktionstastenbelegungen auf das erste von eventuell mehreren Zeichen, so daß der Rest im Puffer verbleibt. Der zweite CALL wertet dagegen direkt den **Tastendruck** aus.

CALL &BB45: KM ARM BREAK
CALL &BB48: KM DISARM BREAK

KM DISARM BREAK sperrt eine Unterbrechung durch ESC, was allerdings nicht im Direktmodus (Eingabemodus) und bei INPUT-Befehlen funktioniert, sondern nur während des Programmablaufs. KM ARM BREAK läßt den Abbruch wieder zu. Wer auch die INPUT-Befehle schützen will, sollte zusätzlich die ESC-Taste umdefinieren, zum Beispiel mit KEY DEF 66,0,126,126,126. Aber Vorsicht: Aus solchen Programmen kommen Sie nur wieder 'lebend' heraus, wenn Sie einen Ausgang vorgesehen haben! Vielleicht sollten Sie die ESC-Taste, die ja jetzt das ASCII-Zeichen 126 liefert, im Programm abfragen und auf diese Weise einen kontrollierten Ausstieg ermöglichen.

CALL &BB4B: KM BREAK EVENT

erzeugt ein 'Unterbrechungs-Ereignis'. In einem BASIC-Programm wird damit ein Druck auf die ESC-Taste simuliert, das heißt, das Programm hält an und wartet bei angeschaltetem Cursor auf einen Tastendruck. ESC führt zum sofortigen Abbruch, eine andere Taste bewirkt eine Fortsetzung. Dieser CALL funktioniert nicht, wenn vorher KM DISARM BREAK aufgerufen wurde.

CALL &BB4E: TXT INITIALIZE

setzt den für die Textausgabe zuständigen Teil des Betriebssystems total zurück. Alle Fenster werden auf den gesamten Bildschirm vergrößert, PEN 1 und PAPER 0 eingestellt, der Cursor in die linke obere Ecke gesetzt, der Transparentmodus und die Textausgabe auf Grafikkoordinaten ausgeschaltet (TAGOFF).

CALL &BB54: TXT VDU ENABLE
CALL &BB57: TXT VDU DISABLE

Mit diesen CALLs lassen sich Textausgaben verhindern bzw. wieder einschalten. Nach 'VDU DISABLE' bleibt jeder PRINT-Befehl wirkungslos, bis 'VDU ENABLE' aufgerufen wird. Auf diese Weise können zum Beispiel auch Floppy-Fehlermeldungen unterdrückt werden.

CALL &BB81: TXT CURSOR ON
CALL &BB84: TXT CURSOR OFF

schalten den Textcursor während des Programmablaufs ein bzw. aus. Das Betriebssystem kontrolliert dabei die Cursorposition, das heißt, nach einem LOCATE-Befehl oder der Ausgabe von Textzeichen wird der Cursor automatisch an die aktuelle Position gesetzt. Es ist deshalb ratsam, den Cursor vor einem PRINT-Befehl abzuschalten, da er sonst während der Textausgabe dazwischenflackert. Mit dem CURSOR-Kommando läßt sich der Cursor auf dem CPC 664/6128 zwar ebenfalls aktivieren; wer jedoch auf die Kompatibilität zum kleinen Bruder CPC 464 Wert legt, sollte lieber die obigen CALLs benutzen.

CALL &BB8A: TXT PLACE CURSOR
CALL &BB8D: TXT REMOVE CURSOR

Diese Routinen dienen ebenfalls dazu, einen Cursor an der aktuellen Textposition erscheinen zu lassen oder zu entfernen. Er unterliegt allerdings nicht der Steuerung des Betriebssystems, das heißt, er bleibt an Ort und Stelle, bis ihn das Programm wieder entfernt. Damit ist es möglich, bei Bedarf eine ganze Legion von Cursoren zu erzeugen (oder wie würden Sie die Mehrzahl bilden?). Da man einen Cursor löscht, indem man einfach einen zweiten Cursor an die gleiche Stelle setzt, sind die obigen beiden CALLs in ihrer Wirkung übrigens vollkommen identisch.

CALL &BB9C: TXT INVERSE

vertauscht die aktuelle PEN- und PAPER-Farbe für die Textausgabe, das heißt, es wird danach invers auf dem Bildschirm geschrieben.

CALL &BB9F,par: TXT SET BACK

Ohne Parameter schaltet dieser CALL den normalen Textaus-

gabemodus ein (Überschreiben), mit einem Parameter (also zum Beispiel CALL &BB9F,1) wird dagegen der Transparentmodus aktiviert.

CALL &BBBA: GRA INITIALIZE

setzt den für Grafikausgaben zuständigen Teil des Betriebssystems zurück: Der Ursprung und der Grafikcursor werden in die linke untere Ecke gesetzt (ORIGIN 0,0: MOVE 0,0), das Grafikfenster auf den gesamten Bildschirm vergrößert und als Schreib- bzw. Hintergrundfarbe GRAPHICS PEN 1 und GRAPHICS PAPER 0 eingestellt.

CALL &BBDE,par,...: GRA SET PEN
CALL &BBE4,par,...: GRA SET PAPER

stellen die Schreib- bzw. Hintergrundfarbe für Grafikausgaben ein, wobei die Anzahl der Parameter die PEN- bzw. PAPER-Nummer bestimmt (zum Beispiel CALL &BBE4,1,1,1 für GRAPHICS PAPER 3). Diese CALLs sind insbesondere auf dem CPC 464 bei Textausgaben auf Grafikcursor-Position (nach TAG) nützlich, da keine entsprechenden BASIC-Einstellbefehle existieren. 664/6128-Besitzer können diese Methode benutzen, wenn sie zum CPC-464-kompatible Programme schreiben wollen.

CALL &BBFF: SCR INITIALIZE

initialisiert den Teil des Betriebssystems, der für die Bildschirm ansteuerung zuständig ist, setzt alle INKs und Farbwechsel auf ihre Standardwerte zurück, schaltet MODE 1 an, löscht den Bildschirm mit PAPER 0 und setzt den Grafik-Schreibmodus auf Überschreiben (Force Mode, siehe SCR ACCESS). Dieser Aufruf ist sehr nützlich, wenn der Bildschirm nach wüsten Farborgien außer Rand und Band ist, an allen Ecken blinkt.

CALL &BC14: SCR CLEAR

löscht den gesamten Bildschirm immer mit PAPER 0, also im Gegensatz zu CLS nicht mit der aktuellen PAPER-Farbe.

CALL &BC4D: SCR HARDWARE ROLL

rollt den gesamten Bildschirm um eine Textzeile nach oben. Die neue Zeile am unteren Rand wird mit PAPER 0 dargestellt.

CALL &BC59,par,...: SCR ACCESS

bestimmt den Schreibmodus für Grafikausgaben (also auch für Text nach TAG). Die Anzahl der Parameter bestimmt den Modus:

0 = Force-Modus (Überschreiben)
1 = XOR-Modus
2 = AND-Modus
3 = OR-Modus

Der Force-Modus ist der Normalzustand. In den anderen Modi werden die Farbwerte der Grafikpunkte mit den vorherigen Farbwerten logisch verknüpft, wodurch sich Spezialeffekte erzielen lassen. Die Einstellung kann man in BASIC auch vornehmen, indem man das Steuerzeichen CHR$(23) mit PRINT abschickt. Das funktioniert allerdings nicht im Grafik-Textmodus (nach TAG), in dem die obigen Modi ja gerade zur Wirkung kommen sollen. Deshalb kann der CALL durchaus nützlich sein, da er es erspart, erst mit TAGOFF die Textausgabe auf Grafikposition abschalten zu müssen.

CALL &BC6E: CAS START MOTOR
CALL &BC71: CAS STOP MOTOR

schalten den Motor des Kassettenrekorders ein bzw. aus. Das funktioniert allerdings nur, wenn Sie vorher die PLAY-Taste gedrückt haben, sonst hört man nur das Relais klicken.

CALL &BCA7: SOUND RESET

bricht die laufenden Sound-Aktivitäten ab und setzt alle Hüllkurven und Tonwarteschlangen zurück.

CALL &BCB6: SOUND HOLD
CALL &BCB9: SOUND CONTINUE

Der erste Aufruf unterbricht alle Sound-Aktivitäten und friert sie quasi ein; der zweite CALL sorgt für die Fortsetzung zu einem späteren Zeitpunkt.

CALL &BD19: MC WAIT FLYBACK

wartet auf den (vertikalen) Strahlrücklauf des Bildschirms (max. 1/50 sec). Dieser CALL kann dazu benutzt werden, um bewegte Grafikobjekte mit dem Bildschirmaufbau zu synchronisieren und eine ruhigere Darstellung zu erreichen. Der Aufruf muß dann allerdings vor **jeder** Bildschirmausgabe erfolgen! Auf dem CPC 664/6128 steht für diesem Zweck der Befehl FRAME zur Verfügung. Wenn das Programm auch auf dem 464 laufen soll, ist jedoch der obige CALL angebracht.

CALL &BD1C,par,...: MC SET MODE

schaltet den MODE um, ohne daß der Bildschirm gelöscht wird, was den Bildschirminhalt stark verfremdet. Die Anzahl der Parameter bestimmt die Modenummer (0,1,2). Da die anderen Teile des Betriebssystems von dieser 'Low-Level-Umschaltung' nichts merken, erfolgen Textausgaben weiterhin so, als wäre noch der ursprüngliche MODE eingeschaltet, sie sind also eventuell grafisch ganz interessant, aber nicht unbedingt lesbar.

CALL &BD37: JUMP RESTORE

setzt alle RAM-Sprungvektoren auf ihre Standardwerte zurück. Dabei handelt es sich exakt um die Einsprungpunkte ins Betriebssystem, die wir hier die ganze Zeit benutzt haben. Die Routine ist eventuell nützlich, wenn ein Maschinenprogramm an den Vektoren gedreht hat, um Betriebssystemaufrufe abzufangen und über eigene Routinen laufen zu lassen. JUMP RESTORE wirft solche Eindringlinge wieder aus dem System heraus (es sei denn, auch die JUMP-RESTORE-Routine wurde gepatcht). Betroffen sind davon leider auch die Floppy-Routinen des CPC, die sich nachträglich in die Kassettenverwaltung einklinken. Es wird also auf Kassette umgeschaltet!

CALL &BD5B,par,...: KL RAM SELECT

Achtung: Nur für den CPC 6128 genießbar! Dieser CALL schaltet die RAM-Speicherbänke um und ermöglicht damit den Zugriff auf die zweiten 64 kByte. Die Anzahl der Parameter bestimmt die RAM-Konfiguration. Für den BASIC-Programmierer kommen zunächst nur zwei Einstellungen ohne Absturzgefahr in Frage: Der Aufruf ohne Parameter (Konfiguration 0) stellt den Normalzustand her, in dem die Speicherbänke 0, 1, 2 und 3 aktiv sind. Jede Bank entspricht 16 kByte RAM. Der Aufruf mit einem Parameter (CALL &BD5B,1 = Konfiguration 1) blendet im Adreßbereich &C000 bis &FFFF (Bildschirmspeicher) anstatt der Bank 3 die Bank 7 ein. Damit wird ein Wechsel zwischen zwei separaten Bildschirmseiten möglich, ohne daß Maschinensprache oder der BANKMAN in Anspruch genommen werden müssen!

Verwendbar sind unter Umständen noch die Konfigurationen Nr. 4 bis 7, die im Adreßbereich &4000 bis &7FFF die Bank 1 durch die Bänke 4...7 ersetzen. Allerdings dürfen sich dann weder Teile eines BASIC-Programms noch Variablen in diesem Bereich herumtreiben, da der BASIC-Interpreter mit Sicherheit sauer reagiert, wenn ihm mit einer spontanen Bankumschaltung sein 'Arbeitsmaterial' entzogen wird. Man muß sich also mit MEMORY &3FFF auf die unteren 16 kByte (Bank 0) beschränken.

(Matthias Uphoff/cd)

Wer suchet, der findet

Suchen & Ersetzen in ASCII-Dateien

Auf den ersten Blick scheint es kaum Vorteile zu bringen, Basic-Programme mit SAVE "name",a als Textdateien abzuspeichern: Sie benötigen in dieser Form mehr Platz auf der Diskette, und der Speicher- beziehungsweise Ladevorgang dauert länger. Trotzdem bietet das ASCII-Format einige Möglichkeiten, die nicht zu verachten sind.

An den eingebauten Basic-Editor haben sich zwar alle CPC-Besitzer mehr oder weniger gewöhnt. Leider fehlen jedoch eine ganze Reihe wichtiger Funktionen, wie zum Beispiel das automatische Suchen und Ersetzen von Programmbefehlen oder Variablennamen.
Eine ASCII-Textdatei läßt sich jedoch auch problemlos mit Hilfe eines Basic-Programms in ein Stringarray einlesen, indem man die Befehle OPENIN, LINE INPUT #9 und CLOSEIN benutzt.
Das Programm wurde konsequent in Basic realisiert, damit Sie es ohne Einschränkungen verändern und an Ihre Bedürfnisse anpassen können. Der Nachteil dieser Methode: CPC-464-Besitzer müssen besonders bei längeren Textdateien damit rechnen, daß der Basic-Interpreter die berüchtigte 'Garbage Collection' veranstaltet, um seine Zeichenketten aufzuräumen, wobei er sich mitunter für einige Zeit totstellt. Die Stringverwaltung des CPC 664/6128 wurde in dieser Hinsicht verbessert, so daß der Effekt kaum noch spürbar ist.
Nach dem Programmstart erscheint das Hauptmenü. Durch Eingabe eines Buchstabens werden die folgenden Optionen ausgewählt:
<L> **Datei laden:** Diesen Punkt werden Sie meistens als ersten anwählen, da die anderen Optionen sonst sinnlos sind. In der unteren Bildschirmhälfte erscheint ein Fenster, in dem Sie den Dateinamen **mit Namenserweiterung** (also zum Beispiel PROG.BAS) eingeben. Er muß wie gesagt eine **ASCII-Datei** bezeichnen, da es sonst einen 'File Type Error' gibt. Nach dem Laden wird der Dateiname im Fenster oben links ständig angezeigt.
<S> **Datei speichern:** Schreibt die Datei wieder auf Kassette/Diskette, nachdem sie mit der Funktion 'Ersetzen' verändert wurde. Wie beim Laden wird der Dateiname in einem Fenster angefordert.
<A> **Datei anzeigen:** In der unteren Bildschirmhälfte erscheint ein Fenster mit der ersten Programm- beziehungsweise Textzeile. Wenn Sie die Leertaste gedrückt halten, wird die Ausgabe fortgesetzt, bis Sie die Taste wieder loslassen oder das Dateiende erreicht ist. Falls der Schalter 'Druckerausgabe' (siehe unten) auf 'ein' steht, wird die gesamte Datei auf dem Drucker ausgegeben, ohne daß Sie die Leertaste niederhalten müssen.
<Z> **Programmende:** Beendet nach einer Sicherheitsabfrage das Programm und veranlaßt die Rückkkehr ins Basic. Ein Abbruch über ESC ist gesperrt.
<F> **Finden:** Listet alle Zeilen auf, die ein bestimmtes Wort oder eine bestimmte Zeichenfolge enthalten. Beeinflußt wird der Suchvorgang durch die Schalter '<U> Unabhängig von Groß-/Kleinschreibung' und '<N> Nur ganze Wörter' (siehe unten). Zunächst werden Sie aufgefordert, den Suchbegriff einzugeben. Wie bei <A> werden danach die betreffenden Zeilen fortlaufend angezeigt, solange Sie die Leertaste drücken. Falls der Schalter 'Druckerausgabe' (siehe unten) auf 'ein' steht, werden dagegen die Zeilen ohne Niederhalten der Leertaste auf dem Drucker protokolliert.
Der Menüpunkt 'Finden' ist bei der Fehlersuche sehr hilfreich: Sie können sich alle Programmzeilen zeigen lassen, in denen ein bestimmter Befehl (zum Beispiel ein Unterprogrammaufruf) oder Variablenname vorkommt.
<E> **Ersetzen:** Diese Option funktioniert im Prinzip genau wie 'Finden', das heißt, zunächst werden Sie aufgefordert, einen Suchbegriff einzugeben. Danach bestimmen Sie durch eine zweite Eingabe eine Zeichenfolge, gegen die der Suchbegriff ausgetauscht werden soll. Von der Stellung des Schalters '<N> Ohne Nachfrage' hängt es ab, ob der Vorgang vollautomatisch abläuft oder vor jedem Austauschvorgang eine Bestätigung (J für ja, N für nein) angefordert wird. Falls der Schalter 'Druckerausgabe' (siehe unten) auf 'ein' steht, werden alle geänderten Zeilen auf dem Drucker protokolliert.
Der Menüpunkt 'Ersetzen' ist zum Beispiel nützlich, wenn Sie in einem BASIC-Programm lange Variablennamen durch kürzere ersetzen wollen, um Speicherplatz zu sparen.
Bei den folgenden vier Menüpunkten handelt es sich um Schalter. Ein Minuszeichen hinter der Option symbolisiert die Stellung 'aus', ein Pluszeichen die Stellung 'ein'. Wenn Sie den Buchstaben drücken, wechselt die Schalterstellung.
<U> **Unabhängig von Groß-/Kleinschreibung:** Bestimmt, ob beim Finden und Ersetzen zwischen Groß- und Kleinschreibung unterschieden wird oder nicht. In der Schalterstellung 'ein' findet das Programm zum Beispiel einen Variablennamen auch dann, wenn Sie ihn uneinheitlich groß und klein geschrieben haben. Bei Texten kann dieser Schalter bewirken, daß ein Wort auch dann erfaßt wird, wenn es zufällig am Satzanfang steht und deshalb groß geschrieben wird.
<W> **Nur ganze Wörter:** Mit der Schalterstellung 'ein' wird beim Finden und Ersetzen ein Begriff nur dann erfaßt, wenn er als eigenständiges Wort auftaucht, also links und rechts durch 'Separatoren' (Leerzeichen, Satzzeichen, Anführungsstriche...) eingerahmt ist. Angenommen, Sie wollen den Variablennamen 'P' suchen und ersetzen. Es wäre natürlich unerfreulich, wenn das Programm an jeder Stelle zuschlägt, an der ein 'P' steht. Wenn jedoch die Option <W> eingeschaltet ist, werden nur Ps erfaßt, die alleine stehen!
<N> **Ohne Nachfrage:** Normalerweise wird vor jedem Ersetzen des Suchbegriffs angehalten, so daß Sie bestätigen oder ablehnen können. Wenn dieser Schalter in der Position 'ein' steht, erfolgt das Ersetzen jedoch vollautomatisch ohne Rückfrage. Diese Option sollte deshalb mit etwas Vorsicht benutzt werden: Sorgen Sie mit den Menüpunkten <U> und <W> dafür, daß es beim Erkennen des Suchbegriffs nicht zu Verwechslungen kommt!
<D> **Druckerausgabe:** Wenn dieser Schalter in der Position 'ein' steht, werden bei den Funktionen 'Anzeigen', 'Finden' und 'Ersetzen' die Bildschirmausgaben auf dem Drucker protokolliert.
Zum Abschluß noch zwei Hinweise für die Anpassung des Programms an Ihren persönlichen Bedarf: Die Zeile 70 bestimmt die maximale Anzahl Zeilen, die das Programm laden kann (Standard 1000). Falls nötig können Sie hier der Variablen maxz einen anderen Wert zuweisen. Der String sep$ in Zeile 90 enthält alle Zeichen, die von der Option '<W> Nur ganze Wörter' als Separatoren erkannt werden (siehe oben). Auch diese Auswahl können Sie beliebig verändern.

(Matthias Uphoff/cd)

für 464-664-6128

```
10 '******* Finden & Ersetzen ******              [835]
20 '***      1989 Matthias Uphoff ***             [2138]
30 '                                              [117]
40 '***** Einige Voreinstellungen                 [1245]
50 MODE 2:INK 0,13:INK 1,0:BORDER 3               [1962]
60 DEFINT a-z                                     [553]
70 maxz=1000                                      [615]
80 DIM z$(maxz)                                   [50]
90 sep$=" '#()*+,-/:;<=>@[\]^"+CHR$(34)           [1761]
100 WINDOW#1,3,78,14,24                           [1055]
110 WINDOW#2,1,80,13,25                           [1309]
120 WINDOW#3,61,79,2,11                           [1164]
130 WIDTH 255                                     [957]
140 CALL &BB48:'Break sperren                     [479]
150 KEY DEF 66,0,126                              [915]
160 OPENOUT"X":MEMORY HIMEM-1:CLOSEOUT            [2858]
170 uflag=0:wflag=0:nflag=0:dflag=0               [2591]
180 '                                             [117]
190 '***** Anzeige Hauptmenue                     [1122]
200 links=1:rechts=21                             [1377]
210 oben=1:unten=12                               [1560]
220 GOSUB 1790:'Rahmen zeichnen                   [1541]
230 LOCATE 3,3:PRINT"Finden & Ersetzen"           [2108]
240 LOCATE 3,4:PRINT"(C) 1989 MU / DMV"           [2388]
250 LOCATE 4,9:PRINT"Aktuelle Datei:"             [2223]
260 a$="":GOSUB 2280                              [1054]
270 links=22:rechts=59                            [876]
280 GOSUB 1790:'Rahmen zeichnen                   [1541]
290 LOCATE 35,1:PRINT" Hauptmenue "               [1686]
300 DATA <L> Datei laden,<S> Datei speiche [11995]
rn,<A> Datei anzeigen,<Z> Programmende,<F>
 Finden,<E> Ersetzen,<U> Unabh. von Gross/
Kleinschr.,<W> Nur ganze Woerter,<N> Erset
zen ohne Nachfrage,<D> Druckerausgabe
310 FOR i=2 TO 11                                 [598]
320   READ a$:LOCATE 24,i:PRINT a$                [945]
330 NEXT                                          [350]
340 GOSUB 1920:'Schalter anzeigen                 [2157]
350 links=60:rechts=80                            [1339]
360 GOSUB 1790:'Rahmen zeichnen                   [1541]
370 '                                             [117]
380 '***** Hauptmenue/Abfrage                     [1059]
390 GOSUB 2220:'Auf Taste warten                  [2550]
400 ON INSTR("LSAZFEUWND",a$) GOSUB 440,64 [4997]
0,830,980,1070,1300,1660,1670,1680,1690
410 GOTO 390                                      [494]
420 '                                             [117]
430 '***** Datei laden                            [696]
440 GOSUB 1720:'Rahmen 1                          [1470]
450 LOCATE 38,17:PRINT" Laden ";                  [1355]
460 GOSUB 2020:'Hinweise 1                        [1015]
470 LOCATE 24,19:INPUT "Dateiname: ",a$           [3004]
480 CLS#3                                         [371]
490 IF a$="" THEN 600                             [803]
500 ERASE z$:DIM z$(maxz)                         [1269]
510 WINDOW SWAP 0,3                               [1029]
520 nz=0:OPENIN a$                                [789]
530 WHILE NOT EOF                                 [1840]
540   nz=nz+1                                     [266]
550   LINE INPUT#9,z$(nz)                         [1802]
560 WEND                                          [390]
570 CLOSEIN                                       [752]
580 CLS:WINDOW SWAP 0,3                           [484]
590 GOSUB 2280:'Dateinamen anz.                   [1875]
600 CLS#2                                         [372]
610 RETURN                                        [555]
620 '                                             [117]
630 '***** Datei speichern                        [801]
640 GOSUB 1720:'Rahmen 1                          [1470]
650 LOCATE 36,17:PRINT" Speichern "               [3062]
660 IF name$="" THEN GOSUB 2140:GOTO 790          [2274]
670 GOSUB 2020:'Hinweise 1                        [1015]
680 LOCATE 24,19:INPUT "Dateiname: ",a$           [3004]
690 CLS#3                                         [371]
700 IF a$="" THEN 790                             [589]
710 WINDOW SWAP 0,3                               [1029]
720 OPENOUT a$                                    [765]
730 FOR i=1 TO nz                                 [383]
740   PRINT#9,z$(i)                               [1007]
750 NEXT                                          [350]
760 CLOSEOUT                                      [902]
770 CLS:WINDOW SWAP 0,3                           [484]
780 GOSUB 2280:'Dateinamen anz.                   [1875]
790 CLS#2                                         [372]
800 RETURN                                        [555]
```

Listing FIND

```
810 '                                              [117]
820 '***** Datei Anzeigen                         [1339]
830 GOSUB 1760:'Rahmen 2                          [1141]
840 LOCATE 37,13:PRINT" Anzeigen "                [2037]
850 IF name$="" THEN GOSUB 2140:GOTO 940          [2276]
860 IF dflag THEN GOSUB 2070 ELSE GOSUB 20 [1373]
60
870 CLS#1                                          [373]
880 FOR i=1 TO nz                                  [383]
890   PRINT#1,z$(i)                                [489]
900   IF dflag THEN PRINT#8,z$(i):GOSUB 22 [2049]
00 ELSE GOSUB 2220
910   IF escflag THEN 940                          [535]
920 NEXT i                                         [375]
930 GOSUB 2110:'Dateiende                          [877]
940 CLS#2:CLS#3                                    [637]
950 RETURN                                         [555]
960 '                                              [117]
970 '***** Programmende                           [1124]
980 GOSUB 1720:'Rahmen 1                          [1470]
990 LOCATE 34,17:PRINT" Programmende "            [2398]
1000 LOCATE 27,19                                  [731]
1010 PRINT"Das Programm beenden (J/N) ?"          [3068]
1020 GOSUB 2220:'Auf Taste warten                 [2550]
1030 IF (a$="N") OR escflag THEN CLS#2:CLS [1311]
#3:RETURN
1040 IF a$<>"J" THEN 1020                         [1246]
1050 MODE 2:CALL &BB00:END                        [1509]
1060 '                                             [117]
1070 '***** Finden                                 [607]
1080 GOSUB 1760:'Rahmen 2                         [1141]
1090 LOCATE 38,13:PRINT" Finden ".                [1357]
1100 IF name$="" THEN GOSUB 2140:GOTO 1260 [2341]
1110 GOSUB 2020:'Hinweise 1                       [1015]
1120 CLS#1:PRINT#1,"***** Suchbegriff: "; [2094]
1130 LINE INPUT#1,such$                           [1574]
1140 CLS#3                                         [371]
1150 IF such$="" THEN 1260                        [1546]
1160 IF uflag THEN such$=UPPER$(such$)            [1830]
1170 IF dflag THEN GOSUB 2070 ELSE GOSUB 2 [1373]
060
1180 FOR i=1 TO nz                                 [383]
1190   p=1:GOSUB 2350:'Suchen                     [2489]
1200   IF p=0 THEN 1240                            [730]
1210   PRINT#1,z$(i)                               [489]
1220   IF dflag THEN PRINT#8,z$(i):GOSUB 2 [2049]
200 ELSE GOSUB 2220
1230   IF escflag THEN 1260                        [649]
1240 NEXT                                          [350]
1250 GOSUB 2110:'Dateiende                         [877]
1260 CLS#2:CLS#3                                   [637]
1270 RETURN                                        [555]
1280 '                                             [117]
1290 '***** Ersetzen                               [402]
1300 GOSUB 1760:'Rahmen 2                         [1141]
1310 LOCATE 37,13:PRINT" Ersetzen "               [2003]
1320 IF name$="" THEN GOSUB 2140:GOTO 1620 [2333]
1330 GOSUB 2020:'Hinweise 1                       [1015]
1340 CLS#1:PRINT#1,"***** Suchbegriff: "; [2094]
1350 LINE INPUT#1,such$                           [1574]
1360 CLS#3                                         [371]
1370 IF such$="" THEN 1620                        [1346]
1380 IF uflag THEN such$=UPPER$(such$)            [1830]
1390 PRINT#1,"***** Ersetzen durch: ";            [2550]
1400 LINE INPUT#1,ers$                            [1643]
1410 GOSUB 2070:'Hinweise 2                       [2537]
1420 FOR i=1 TO nz                                 [383]
1430   f=0                                         [392]
1440   p=1:GOSUB 2350:'Suchen                     [2489]
1450   IF p=0 THEN 1590                            [772]
1460   IF nflag THEN 1530                          [573]
1470   PRINT#1,LEFT$(z$(i),p-1)+CHR$(24)+M [6333]
ID$(z$(i),p,LEN(such$))+CHR$(24)+MID$(z$(i
),p+LEN(such$))
1480   PRINT#1,"***** Ersetzen durch ";CHR [3991]
$(24);ers$;CHR$(24);" (J/N)?"
1490   GOSUB 2220:'Auf Taste warten               [2550]
1500   IF escflag THEN 1620                        [704]
1510   IF a$="N" THEN p=p+1:GOTO 1550             [1102]
1520   IF a$<>"J" THEN 1490                       [1208]
1530   z$(i)=LEFT$(z$(i),p-1)+ers$+MID$(z$ [3559]
(i),p+LEN(such$))
1540   p=p+LEN(ers$):f=-1                         [1544]
1550   GOSUB 2350:'Suchen                         [2137]
1560   IF p>0 THEN 1460                            [996]
1570   IF f THEN PRINT#1,z$(i)                    [1075]
1580   IF dflag AND f THEN PRINT#8,z$(i)          [1593]
1590   GOSUB 2200:IF escflag THEN 1620            [2023]
```

Listing FIND

```
1600 NEXT i                                        [375]
1610 GOSUB 2110:'Dateiende                         [877]
1620 CLS#2:CLS#3                                   [637]
1630 RETURN                                        [555]
1640 '                                             [117]
1650 '***** Flags umschalten                      [1697]
1660 uflag=NOT uflag:GOSUB 1920:RETURN            [1816]
1670 wflag=NOT wflag:GOSUB 1920:RETURN            [3811]
1680 nflag=NOT nflag:GOSUB 1920:RETURN            [2546]
1690 dflag=NOT dflag:GOSUB 1920:RETURN            [3056]
1700 '                                             [117]
1710 '***** Rahmen 1                               [454]
1720 links=22:rechts=59                            [876]
1730 oben=17:unten=21                             [2249]
1740 GOTO 1790                                     [341]
1750 '***** Rahmen 2                               [453]
1760 links=1:rechts=80                            [1414]
1770 oben=13:unten=25                             [1438]
1780 '***** Rahmen zeichnen                        [451]
1790 LOCATE links,oben:PRINT CHR$(150);           [2798]
1800 PRINT STRING$(rechts-links-1,154);           [2142]
1810 PRINT CHR$(156);                             [1037]
1820 FOR i=oben+1 TO unten-1                      [1507]
1830   LOCATE links,i:PRINT CHR$(149);            [2586]
1840   LOCATE rechts,i:PRINT CHR$(149);           [3718]
1850 NEXT                                          [350]
1860 LOCATE links,unten:PRINT CHR$(147);          [2772]
1870 PRINT STRING$(rechts-links-1,154);           [2142]
1880 PRINT CHR$(153);                              [970]
1890 RETURN                                        [555]
1900 '                                             [117]
1910 '***** Flagstatus anzeigen                   [2564]
1920 i=8                                           [412]
1930 f=uflag:GOSUB 1970                            [724]
1940 f=wflag:GOSUB 1970                            [892]
1950 f=nflag:GOSUB 1970                           [1242]
1960 f=dflag                                       [684]
1970 LOCATE 57,i:i=i+1                            [1742]
1980 IF f THEN PRINT "+" ELSE PRINT "-"           [1049]
1990 RETURN                                        [555]
2000 '                                             [117]
2010 '***** Bedienungshinw. 1                     [2256]
2020 LOCATE 64,6:PRINT"Abbruch durch"             [1809]
2030 LOCATE 64,7:PRINT"leere Eingabe"             [2656]
2040 RETURN                                        [555]
2050 '***** Bedienungshinw. 2                     [2251]
2060 LOCATE 63,6:PRINT"Weiter mit SPACE"          [2167]
2070 LOCATE 63,7:PRINT"Abbruch mit ESC"           [2198]
2080 RETURN                                        [555]
2090 '                                             [117]
2100 '***** Dateiende                              [911]
2110 PRINT#1,CHR$(7);"***** Dateiende erre [3261]
icht: [ESC] druecken"
2120 GOTO 2160                                     [363]
2130 '***** Fehlermeldung                          [440]
2140 LOCATE 24,19                                  [590]
2150 PRINT CHR$(7);"Keine Datei geladen: [ [3305]
ESC] druecken"
2160 GOSUB 2220: IF NOT escflag THEN 2160  [4000]
2170 RETURN                                        [555]
2180 '                                             [117]
2190 '***** Test auf ESC                           [696]
2200 a$=INKEY$:GOTO 2240                           [693]
2210 '***** Auf Taste warten                      [1782]
2220 WHILE INKEY$<>"":WEND                        [1786]
2230 a$=UPPER$(INKEY$):IF a$="" THEN 2230         [1890]
2240 escflag = (a$=CHR$(126))                      [693]
2250 RETURN                                        [555]
2260 '                                             [117]
2270 '***** Dateiname setzen                      [1379]
2280 name$=UPPER$(a$)                              [293]
2290 LOCATE 4,10:PRINT STRING$(15,32);            [1170]
2300 LOCATE 6,10                                   [496]
2310 IF name$="" THEN PRINT "-- Keine --" [2306]
ELSE PRINT name$
2320 RETURN                                        [555]
2330 '                                             [117]
2340 '***** Suchen                                 [335]
2350 IF uflag THEN a$=UPFER$(z$(i)) ELSE a [1841]
$=z$(i)
2360 p=INSTR(p,a$,such$)                          [1140]
2370 IF (NOT wflag) OR p=0 THEN RETURN            [1009]
2380 IF p=1 THEN 2400                             [1017]
2390 IF INSTR(sep$,MID$(a$,p-1,1))=0 THEN  [2389]
2420
2400 IF p+LEN(such$)-1=LEN(a$) THEN RETURN [2880]
2410 IF INSTR(sep$,MID$(a$,p+LEN(such$),1) [4421]
) > 0 THEN RETURN
2420 p=p+1:GOTO 2360                              [1517]
```

Listing FIND

2000 plus ROM gleich 3000

Die wundersame Wandlung eines Amstrad-Druckers

Der DMP 2000 ist einer der am weitest verbreiteten Drucker für den CPC, seinen Bruder DMP 2160 nicht mitgerechnet. Leider fehlt ihm so einiges an Funktion und Möglichkeiten, die der Nachfolger DMP 3000/3160 mit sich brachte, darunter der IBM-Zeichensatz und ein erweitertes RAM. Aber nicht verzweifeln, mit ein paar Werkzeugen, gutem Willen und einigen Bauteilen läßt sich der gute alte Nadeldrucker in einen guten (fast) neuen Nadeldrucker verwandeln.

Zuerst ein Hinweis: Auch wenn der DMP 2000 nicht der Neueste ist, das Öffnen des Gerätes läßt eventuell noch vorhandene Garantieansprüche erlöschen, es sollte sich also jeder Bastler über diesen Punkt klar sein. Außerdem ist unbedingt der Netzstecker zu ziehen, so mancher Möchtegern-Elektroniker hat sich auf diese Weise schon ein paar stehende Haarspitzen geholt, ein nicht auf die leichte Schulter zu nehmendes Problem. So, jetzt geht's aber los:

Der Umbau

Der Umbau gliedert sich in zwei Abschnitte: Dem Austausch des ROMs und dem Ausbau des Pufferspeichers von 2 auf 8 kByte. Beide Schritte können auch jeweils ohne den anderen durchgeführt werden.

1. Austausch des ROMs
Als erstes benötigen Sie ein ROM des DMP 3000. Dieses hat eine Speicherkapazität von 32 kByte, während im DMP 2000 ein 16-kByte-EPROM steckt. Daher sind für den Umbau Änderungen an der Platine nicht zu vermeiden.

Nachdem Sie den Netzstecker des Druckers aus der Steckdose gezogen haben, öffnen Sie das Gehäuse, indem Sie die sechs Schrauben auf der Vorderseite lösen und das Oberteil vorsichtig abheben. Vorher muß noch das Farbband entfernt werden.

Auf der Platine sind zwei ICs gesockelt. Das größere ist das EPROM; es ist auch an einem Aufkleber zu erkennen. Heben Sie es vorsichtig mit einem Schraubenzieher oder ähnlichem ab. Dann lösen Sie die beiden Schrauben, mit denen die Platine befestigt ist und heben die Leiterplatte aus den Befestigungen heraus. Die vielen Steckverbindungen brauchen nicht gelöst zu werden.

Jetzt suchen Sie drei Lötpunkte oberhalb des EPROM-Steckplatzes, die mit J1 gekennzeichnet sind. Zwei Lötpunkte sind auf der Rückseite der Platine miteinander verbunden. Mit einer scharfen Klinge trennen Sie diese beiden Punkte und verbinden statt dessen den mittleren mit dem vorher unbenutzten. An den jetzt unbenutzten Lötpunkt löten Sie ein Kabel, das bis zum Pin 27 (A 14) des ROM-Sockels reicht. Dieser Pin 27 ist mit Pin 28 und damit mit +5 Volt verbunden. Diese Verbindung müssen Sie ebenfalls durchtrennen. Als nächstes löten Sie das vorher angelötete Kabel an Pin 27 an. Jetzt kann das DMP 3000 ROM mit der Einkerbung in Richtung Centronicsbuchse in den Sockel eingesetzt werden.

2. Erweiterung des Druckerpuffers
Der Austausch des RAMs ist weniger aufwendig. Hierzu brauchen Sie ein statisches 8-kByte-RAM mit der Bezeichnung 6264. Den 24-poligen RAM-Baustein im Drucker mit der Bezeichnung 6116 erkennen Sie daran, daß er in einem zu großen Sockel steckt. Zunächst suchen Sie wieder drei Lötpunkte, die mit J3 gekennzeichnet sind. Dann ändern Sie die Verbindung des mittleren Punktes wie oben bei dem ROM um. Jetzt kann das neue RAM gegen das alte ausgetauscht werden. Vor der Centronicsbuchse befindet sich ein Jumper, den Sie noch umstecken sollten. Dadurch wird das AFD-Signal an Pin 14 der Buchse geschaltet. Es bewirkt einen Zeilenvorschub, falls ein Low-Pegel anliegt. Die CPCs machen davon keinen Gebrauch. Jetzt brauchen Sie nur noch den Drucker wieder zusammenzuschrauben und die Arbeit ist getan. Nach dem fertigen Umbau können Sie Ihre Arbeit prüfen. Schalten Sie den Drucker in den Epson-Modus. Dann können Sie kursiv in NLQ drucken:

```
PRINT#8,CHR$(27)"m"CHR$(0)CHR$(27)"x"CHR$(1)CHR$(27)"4";
"NLQ-kursiv drucken mit einem DMP 3000 ROM"
```

Der DMP 3000 besitzt ein neues Steuerzeichen, das den Drucker auf die gewünschte Kompatibilität einstellt: "ESC m <n>". <n> steht für eine Zahl von 0 bis 2: 0: kennzeichnet den EPSON FX-Modus 1: steht für dem IBM-Zeichensatz Nr. 1 und 2: für den IBM-Zeichensatz Nr. 2.

Bei den DIP-Schaltern hat sich die Bedeutung von DS1-7 und DS1-8 geändert. Jetzt wird damit der Standardzeichensatz ausgewählt:

Zeichensatz	*DS1-7*	*DS1-8*
Epson FX-Standard	*OFF*	*OFF*
Epson FX-NLQ	*ON*	*OFF*
IBM #1	*OFF*	*ON*
IBM #2	*ON*	*ON*

Ein weiterer Steuercode, den zwar auch der DMP 2000 versteht, jedoch den meisten Besitzern dieses Druckers unbekannt sein wird, ist: "ESC ! <n>" <n> ist eine Bit-signifikante Zahl, mit der man, mit Hilfe eines gesetzten Bits, die Schriftart einstellen kann.

Bit 0: Elite
1: immer 0
2: Schmalschrift
3: Fettdruck
4: Doppeldruck
5: Doppelte Breite
6: immer 0
7: Kursivdruck

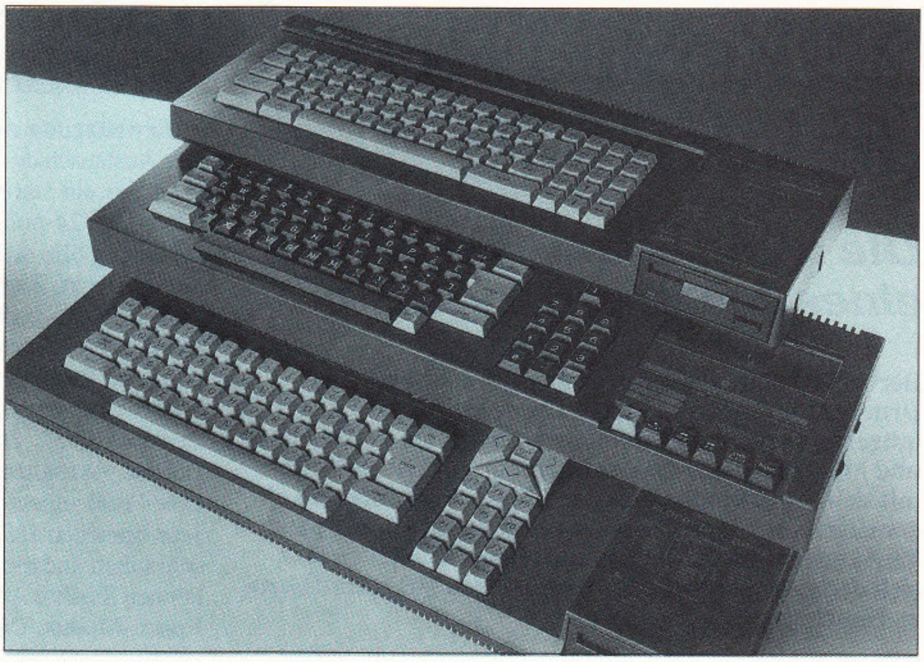
Die Standard-Schrift wird mit n=0 gewählt. Auch gibt es beim DMP 2000 die Möglichkeit, daß der Drucker alle ASCII-Werte der ankommenden Daten hexadezimal ausgibt. Dazu müssen die Tasten LF und FF beim Einschalten gedrückt werden. Als Test können Sie dieses Programm eingeben:

```
10 FOR i=0 to &7F
20 PRINT#8,CHR$(i);
30 NEXT
```

Zum Schluß noch ein Tip für CPC-Besitzer: Da Ihr Computer nur über eine Sieben-Bit-Schnittstelle verfügt, müssen Sie das achte Bit mittels Software einstellen, um die Grafikzeichen auf das Papier zu bekommen. Die dazu nötigen Steuercodes stehen im Druckerhandbuch.

(Michael Kruse/jb)

Der CPC-664/ 6128-Emulator für den CPC 464



BASIC 1.1 auf AMSTRADs Kleinsten

Viele der Programme, die wir im Laufe der Zeit veröffentlicht haben, haben einen kleinen Mangel, den die Besitzer eines CPC 6128 oder 664 überhaupt nicht bemerken, Besitzer eines CPC 464 allerdings um so mehr, denn auf ihrem Computer sind diese Programme kaum oder sogar überhaupt nicht zum Laufen zu bekommen.

Woran liegt das, wird sich so mancher frischgebackene CPC-Anwender fragen. Nun, dies ist ziemlich einfach zu erklären: BASIC ist Schuld!
Während der Kleinste, der 464, mit dem Locomotive-BASIC 1.0 ausgeliefert wurde, bekamen seine Nachfolger 664/6128 eine verbesserte Version 1.1 ins ROM gesetzt. Der Erfolg: Das um neue Befehle erweiterte BASIC 1.1 wird nicht hundertprozentig vom 464 akzeptiert.

Keine Möglichkeit oder etwa doch...?

Wir haben für den Fall, daß Sie Besitzer des CPC 464 sind und auch die anderen Programme nutzen wollen, eine, wie wir glauben, sehr gute Lösung parat: den EMULATOR.
Er ist ein Maschinenprogramm, welches den CPC 464, oder besser sein BASIC, um die erweiterten Befehle des BASIC 1.1 bereichert. Dabei werden die Befehle nicht als RSXe eingebunden, sondern als 'echte' Befehle, somit können Programme der 'großen Brüder' fast ohne Probleme ablaufen. Das 'fast' steht für einen ganz geringen Prozentsatz, der sich nicht sofort zu erkennen gibt. Hier kommt es aufs Probieren an.

Die Befehle

CLEAR INPUT dient zum Löschen des Tastaturpuffers.
COPYCHR$(# WINDOW) liest ein Zeichen an der angegebenen Cursor-Position im angegebenen Textfenster.
CURSOR Schalter1,Schalter2 ermöglicht eine Ab- oder Anschaltung des Cursors im Programmlauf.
DEC$(numerischer Ausdruck, Formatstring) ist eigentlich auch schon auf dem CPC 464 vorhanden, hierbei jedoch ist der Interpreterfehler ausgemerzt, der nach DEC$ zwei geschlossene Klammern fordert.
FILL Farbstift ist eine schnelle Füllroutine, die an der aktuellen Position des Grafikcursors beginnt, also mit MOVE x,y bestimmt werden muß.
FRAME ist ein Synchronisierungsbefehl für den Strahlrücklauf, damit wird das Scrolling sanfter.
GRAPHICS PAPER Farbstift wird benötigt, um die Farbe des Grafikhintergrundes bei Textausgabe auf der Grafikposition und beim MASK-Befehl zu bestimmen.
GRAPHICS PEN Farbstift, Hintergrund wählt die Schreibfarbe und den Hintergrund für eine Grafik aus, wobei der zweite Parameter für folgendes gilt:
– 0 = normal überschreibend
– 1 = transparent.

MASK Bitmaske erstellt anstatt der durchgehenden Linien unterbrochene, wobei der Wert 'Bitmaske' (eine Zahl zwischen 0 und 255) die Strichelung bestimmt.

PEN # WINDOW, Farbstift, Hintergrund, ebenfalls vom CPC 464 bekannt, nur kann hierbei mit dem letzten Parameter bestimmt werden, ob der Hintergrund transparent (1) oder normal (0) erscheinen soll.
MERGE und CHAIN MERGE"Dateiname" diesmal auch in einwandfreiem Zustand beim Diskettenbetrieb.

Die Parameterübergabe bei CALL und RSX kann jetzt direkt vorgenommen werden, hier ein Beispiel:
Statt a$="*.BAK":|ERA,@a$
jetzt |ERA,"*.BAK"

Noch ein paar Hinweise

Sollte ein CPC-664/6128-Programm auf Ihrem CPC 464 einmal mit 'SYNTAX ERROR' abbrechen, so editieren Sie die Zeile und drücken sofort die ENTER-Taste, danach sollte die Zeile übernommen werden.

Tippen Sie zuerst den Lader für das Maschinenprogramm ab, und speichern Sie ihn auf Datenträger, danach starten Sie ihn mit RUN, er erzeugt das Binärfile. Jetzt noch das BASIC-Programm abtippen, abspeichern und mit RUN starten, es lädt den Binärteil nach.

Kassettenbesitzer sollten zwei Kassetten bereithalten. Auf der ersten Kassette speichern Sie Listing 2, dann tippen Sie Listing 1 ab und speichern es auf der zweiten Kassette ab. Danach die erste wieder einlegen und Listing 1 mit RUN starten, das Binärfile wird nach dem BASIC-Teil auf die Kassette geschrieben. Wenn Sie jetzt zurückspulen, können Sie den Emulator mit RUN"EMU" starten.

(Matthias Uphoff/jb)

für 464

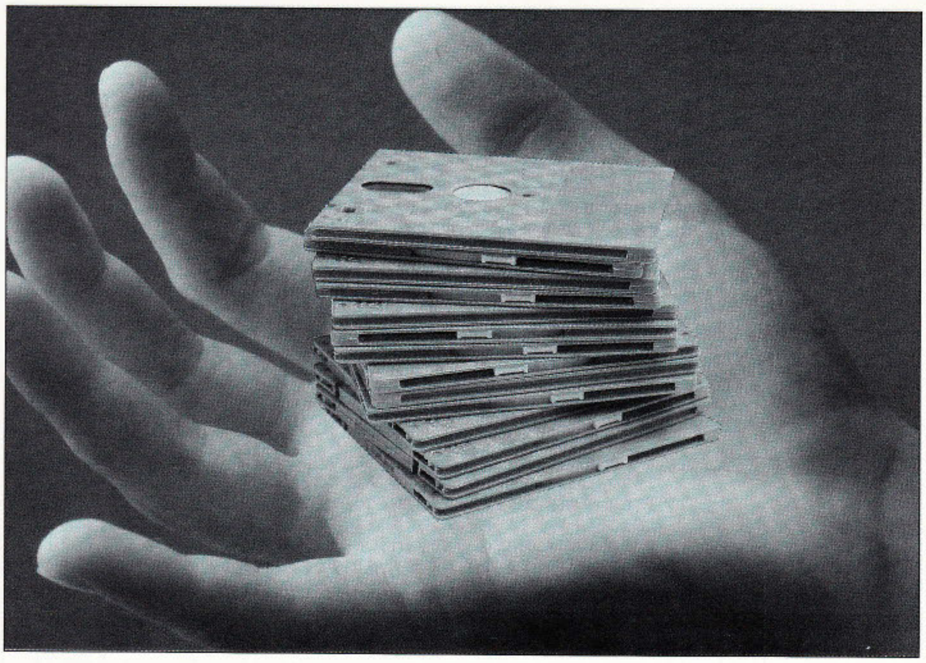
```
5 'Lader fuer 664/6128-Emulator          [1585]
10 SYMBOL AFTER 256                      [1408]
20 MEMORY &A1FF                          [450]
30 SYMBOL AFTER 240                      [1392]
40 LOAD"emu.bin"                         [1942]
50 CALL &A200                            [597]
60 PRINT"BASIC 1.1 installiert"          [1931]
```

```
1000 ' Generated by DataGen 1.0           [2863]
1010 MEMORY &A1FF                         [450]
1020 FOR adr=&A200 TO &A639 STEP 8        [1122]
1030   FOR i=adr TO adr+ 7                [1011]
1040     READ b$                          [315]
1050     byte=VAL("&"+b$)                 [465]
1060     POKE i,byte                      [294]
1070   NEXT i                             [375]
1080 NEXT adr                             [547]
1085 SAVE"emu.bin",b,&A200,&439           [1427]
1090 END                                  [110]
1100 DATA 3E,C3,32,16,AC,21,83,A2         [1312]
1110 DATA 22,17,AC,32,19,AC,21,8E         [1277]
1120 DATA A2,22,1A,AC,32,07,AC,21         [1313]
1130 DATA 9A,A2,22,08,AC,32,04,AC         [1745]
1140 DATA 21,B4,A4,22,05,AC,32,0D         [1184]
1150 DATA AC,21,2A,A3,22,0E,AC,32         [1627]
1160 DATA 01,AC,21,AB,A2,22,02,AC         [1019]
1170 DATA 32,90,BB,21,D8,A2,22,91         [1575]
1180 DATA BB,32,DE,BB,21,11,A3,22         [1810]
1190 DATA DF,BB,32,C0,BB,21,F5,A2         [1232]
1200 DATA 22,C1,BB,32,C3,BB,21,FA         [1543]
1210 DATA A2,22,C4,BB,21,53,A3,22         [1537]
1220 DATA E3,BD,21,62,A3,22,E9,BD         [2104]
1230 DATA 3A,80,BC,FE,C3,C8,32,36         [1770]
1240 DATA A6,2A,81,BC,22,37,A6,3E         [944]
1250 DATA C3,32,80,BC,21,B2,A2,22         [1562]
1260 DATA 81,BC,C9,11,E1,A5,CD,27         [1497]
1270 DATA E3,D0,F1,C3,61,DF,21,E1         [1285]
1280 DATA A5,CD,13,E3,D0,7E,23,C1         [1633]
1290 DATA C1,C9,FE,C5,D0,D6,BA,D1         [751]
1300 DATA EB,4F,06,00,21,1C,A6,09         [1413]
1310 DATA C3,BB,DD,21,91,B2,75,C3         [1126]
1320 DATA 68,A2,E5,3A,36,A6,32,80         [1430]
1330 DATA BC,2A,37,A6,22,81,BC,CD         [1480]
1340 DATA 80,BC,21,80,BC,36,C3,21         [1421]
1350 DATA B2,A2,22,81,BC,E1,D8,C8         [1269]
1360 DATA FE,1A,37,3F,C0,B7,37,C9         [1051]
1370 DATA CD,D5,A5,F3,D9,CB,59,D9         [1082]
1380 DATA FB,C0,E1,E3,CD,55,DD,30         [1455]
1390 DATA 0A,3E,02,CD,FB,C1,E5,CD         [2199]
1400 DATA 9F,BB,E1,E3,E9,CD,DB,A5         [1355]
1410 DATA 18,03,CD,DE,A5,F3,D9,CB         [1527]
1420 DATA 59,D9,FB,C0,E1,E3,CD,55         [1531]
1430 DATA DD,DC,0E,A3,E3,E9,CD,4B         [2240]
1440 DATA C2,CD,D8,A5,F3,D9,CB,59         [1024]
1450 DATA D9,FB,C0,CD,55,DD,D0,3E         [1468]
1460 DATA 04,CD,FB,C1,E5,CD,59,BC         [1153]
1470 DATA E1,C9,2B,7E,FE,20,28,FA         [1225]
1480 DATA FE,2C,C0,2B,E5,2A,34,AE         [1763]
1490 DATA CD,3F,DD,21,28,A6,CD,AA         [1804]
1500 DATA FF,E1,38,08,FE,BB,C0,CD         [1770]
1510 DATA 93,BB,18,03,CD,E1,BB,D1         [754]
1520 DATA C3,4F,D0,3A,2F,A6,2F,32         [1266]
1530 DATA 33,A6,CD,3C,18,AF,32,33         [714]
1540 DATA A6,C9,3A,33,A6,B7,20,0F         [921]
1550 DATA 3A,39,B3,B8,C2,CC,B1,3A         [605]
1560 DATA 30,A6,B7,CA,CC,B1,C9,3A         [1876]
1570 DATA 46,B3,B7,3A,2F,A6,20,11         [1515]
1580 DATA 07,32,2F,A6,3A,38,B3,47         [1267]
1590 DATA DA,CC,B1,3A,39,B3,47,18         [1282]
1600 DATA DE,E5,D5,C5,21,CF,B1,5F         [1766]
```

Listing Emulator

```
1610 DATA 3A,07,B2,ED,44,47,AF,CB         [1472]
1620 DATA 01,30,05,CB,03,30,01,B6         [2015]
1630 DATA 23,10,F4,4F,7B,32,2F,A6         [960]
1640 DATA 3A,30,A6,B7,20,0A,2A,38         [1225]
1650 DATA B3,7D,AC,A1,AC,C1,18,01         [1524]
1660 DATA F1,47,D1,E1,C3,CC,B1,CA         [1180]
1670 DATA A3,FE,CD,86,0C,32,3A,B3         [1271]
1680 DATA CD,1A,16,CD,FF,16,D2,68         [1264]
1690 DATA A4,E5,FD,E1,D5,DD,E1,CD         [1183]
1700 DATA A9,0B,3A,3A,B3,57,AE,A1         [1195]
1710 DATA 28,7E,3A,38,B3,5F,AE,A1         [1188]
1720 DATA 28,76,06,00,FD,23,CD,2D         [458]
1730 DATA 0C,3A,34,B3,FD,95,38,52         [1326]
1740 DATA 7A,AE,A1,28,4D,7B,AE,A1         [2160]
1750 DATA 20,EA,18,46,AE,77,CB,20         [1201]
1760 DATA E5,CB,01,DC,05,0C,7A,AE         [1551]
1770 DATA A1,28,12,7B,AE,A1,28,0D         [1077]
1780 DATA CB,C0,CB,48,20,07,DD,2B         [1427]
1790 DATA DF,8A,A4,DD,23,CB,A0,CB         [1782]
1800 DATA 09,E1,E5,CB,09,DC,F9,0B         [984]
1810 DATA 7A,AE,A1,28,12,7B,AE,A1         [1767]
1820 DATA 28,0D,CB,E0,CB,68,20,07         [733]
1830 DATA DD,23,DF,8A,A4,DD,2B,CB         [1367]
1840 DATA 01,E1,3A,36,B3,FD,95,30         [1917]
1850 DATA 0F,FD,2B,CD,13,0C,7B,AE         [1894]
1860 DATA A1,28,05,7A,AE,A1,20,A4         [1286]
1870 DATA 2A,34,A6,7D,B4,C8,2B,22         [1518]
1880 DATA 34,A6,DF,78,A4,C3,D3,A3         [1234]
1890 DATA 7B,A4,FD,2A,89,AE,2B,56         [2087]
1900 DATA 2B,5E,2B,22,89,AE,6E,26         [1822]
1910 DATA 00,C9,8D,A4,FD,C5,D5,2A         [466]
1920 DATA 89,AE,01,03,00,CD,18,F6         [1345]
1930 DATA 38,17,EB,22,89,AE,DD,E5         [847]
1940 DATA D1,FD,E5,C1,2B,72,2B,73         [1147]
1950 DATA 2B,71,2A,34,A6,23,22,34         [1282]
1960 DATA A6,D1,C1,C9,FE,03,28,41         [1373]
1970 DATA FE,AB,28,57,FE,9F,CA,4A         [1386]
1980 DATA A5,FE,2C,28,03,FE,28,C0         [1562]
1990 DATA 3E,02,B9,20,64,F1,C1,F5         [2185]
2000 DATA 3E,F8,B8,C0,79,FE,EE,28         [1224]
2010 DATA 15,FE,F5,C0,7E,2B,FE,72         [1151]
2020 DATA 20,FA,7E,3C,20,F6,23,CD         [1430]
2030 DATA 3F,DD,CD,37,DD,28,F1,CD         [999]
2040 DATA FB,CE,CD,37,DD,2C,C3,F5         [1746]
2050 DATA F8,3E,0D,BB,C0,2A,34,AE         [1472]
2060 DATA CD,3F,DD,FE,7C,28,03,FE         [1396]
2070 DATA 83,C0,E1,2A,C2,B0,CD,21         [556]
2080 DATA FB,37,C9,3E,05,BB,C0,3E         [1681]
2090 DATA 1A,91,C0,B8,C0,F1,CD,F5         [1012]
2100 DATA C1,CD,D3,C1,CD,37,DD,29         [526]
2110 DATA CD,60,BB,E5,CD,19,FA,E1         [1119]
2120 DATA C9,3E,49,B9,C0,F1,C1,F5         [872]
2130 DATA 3E,D0,B8,C0,3E,93,B9,C0         [1553]
2140 DATA F1,E5,00,00,AF,CD,0A,FF         [1716]
2150 DATA E1,C9,F1,C1,F5,3E,C8,B8         [1410]
2160 DATA C0,3E,D6,B9,C0,3E,8B,BE         [998]
2170 DATA C0,F1,11,61,A5,1B,C3,DA         [846]
2180 DATA C8,06,00,FF,FF,C9,00,CD         [2264]
2190 DATA 55,DD,38,10,3E,02,CD,FB         [1856]
2200 DATA C1,B7,CC,84,BB,C4,81,BB         [393]
2210 DATA CD,55,DD,D0,3E,02,CD,FB         [1575]
2220 DATA C1,B7,CA,7E,BB,C3,7B,BB         [668]
2230 DATA 23,01,E4,BB,FE,BA,CA,18         [2158]
2240 DATA C2,FE,BB,C2,C6,DD,CD,55         [917]
2250 DATA DD,38,0A,CD,4B,C2,CD,D8         [599]
2260 DATA A5,CD,55,DD,D0,3E,02,CD         [2270]
2270 DATA FB,C1,32,30,A6,C9,CD,4B         [889]
2280 DATA C2,E5,DF,C7,A3,E1,C9,CD         [950]
2290 DATA 55,DD,38,0A,CD,67,CE,32         [1889]
2300 DATA 2F,A6,CD,55,DD,D0,3E,02         [1892]
2310 DATA CD,FB,C1,32,31,A6,C9,CD         [2316]
2320 DATA 09,BB,38,FB,C9,CF,A9,92         [1262]
2330 DATA CF,F6,97,CF,F4,95,CF,F1         [723]
2340 DATA 95,46,49,4C,CC,DD,47,52         [1078]
2350 DATA 41,50,48,49,43,D3,DE,4D         [895]
2360 DATA 41,53,CB,DF,46,52,41,4D         [1794]
2370 DATA C5,E0,43,55,52,53,4F,D2         [1101]
2380 DATA E1,43,4C,45,41,52,20,49         [1288]
2390 DATA 4E,50,55,D4,E2,43,4F,50         [1622]
2400 DATA 59,43,48,52,A4,7E,44,45         [1573]
2410 DATA 52,D2,49,00,AE,A5,88,A5         [952]
2420 DATA B7,A5,19,BD,67,A5,CF,A5         [1232]
2430 DATA 94,95,AE,AF,BC,BD,00,FF         [1309]
2440 DATA 00,FF,FF,00,00,00,00,00         [1953]
2450 DATA 00,00,00,00,00,00,00,00         [1005]
```

Listing Emulator



Klein, aber oho!

Erstaunliches aus einem kByte

Wer will schon immer zwanzig Seiten staubkorngroße Listings eingegeben? Lange Werke sind derweilen des Inhalts Feind, oder wie schon Shakespeare philosophierte: "Kürze ist des Witzes Seele".
Befassen wir uns also einmal mehr mit den Programmen, die durch ihre Präzision und Kompaktheit bestechen. Was dabei zuweilen zustande kommt, ist in größtem Maße erstaunlich, wie die folgenden Kurzprogramme eindrucksvoll belegen.

Programm 1: BIG

Problemstellung: Sie wollen dem Finanzamt die Antwort auf die Frage entlocken, ob Sie nicht die ein oder andere Mark am Jahresende zurückerstattet bekommen. Sie wissen, daß die Herren und Damen auf so manchem Ohre taub sind und beschließen, die Frage in Form eines Banners zu präsentieren, wobei jeder einzelne Buchstabe aus lauter Fragezeichen gebildet wird.

BIG hilft Ihnen dabei. Sie starten das Programm und werden daraufhin aufgefordert, einige Parameter einzugeben. Zuerst wird eine Zeichenkette abgefragt. Um beim Finanzamt-Beispiel zu bleiben: die Kette könnte lauten: "Krieg' ich Geld oder nicht?"

Die nächsten Werte wären x- und y-Streckung, die die Größe der Bannerzeichen darstellen. Der x-Wert kann dabei ins Unermeßliche reichen, der y-Wert sollte nicht über 9 gehen, da es sonst Schwierigkeiten mit der Größe des Papiers geben wird. Das vierte Parameter ist die Nummer des Zeichens, das als Füllung der Zeichenkette dienen soll. In unserem Beispiel wäre ein Fragezeichen angebracht. Fertig! Wenn nun der Drucker eingeschaltet ist, erhalten wir einen riesigen Ausdruck, der auffällig genug ist, um Ihr Anliegen deutlich zu machen.

(Michael Emmer/jf/cd)

Programm 2: Schriftvergrößerung

Bisher mußten Titelscreens mit großem Aufwand erstellt werden und wenn sie auch nur bildschirmgroße Worte darstellen sollten.

Wenn Sie ihr neuestes Programm mit einem solchen relativ einfachen Titelbild schmücken wollen, dann hilft Ihnen das zweite Beispiel kurzer Programmierkunst.

Sie können hiermit zwei maximal sieben Zeichen umfassende Ketten so vergrößern, daß beide nach Erstellen den gesamten Bildschirm einnehmen.

Was ist zu tun? Laden Sie das Programm, sorgen Sie dafür, daß sich eine Datendiskette im Laufwerk befindet und geben Sie die beiden Zeichenketten ein. Das Programm errichtet nun den Bildschirm und, wenn es mit Schreiben fertig ist, sichert es ihn auf Diskette als normale 17 kByte große Binärdatei. Diese kann nach herkömmlicher Weise in den Bildschirmspeicher geladen werden: LOAD "-Name-.bin",&C000

Das Programm arbeitet in MODE 2; das heißt, daß die fertige Datei, will man sie originalgetreu zurückholen, auch in MODE 2 geladen werden muß. Verändert man den MODE vor dem Ladevorgang, so ergeben sich interessante Effekte. In MODE 1 erhalten die Buchstaben Schatten, in MODE 0 erscheinen sehenswerte Effekte, die durch gezielte Farbwahl in interessanter Weise beeinflußt werden können.

(Tobias Sander/jf/cd)

Programm 3: Calculat

Wer sich mathematisch betätigt, stößt irgendwann auf jenen ungeliebten Zweig dieser oft gehaßten Wissenschaft: Bruchrechnung! Alles wäre so einfach, wenn es simple Methoden gäbe, die in den Zahlen verborgenen Werte kgV (kleinstes gemeinsames Vielfache) und ggT (größter gemeinsamer Teiler) zu bestimmen. Hat man die erst einmal berechnet, dann, ja dann ist Bruchrechnung kein Problem mehr.

Der CPC kann's kurz und knapp mit weniger als 1000 Bytes. 19 Zeilen reines BASIC-Listing helfen Ihnen ab sofort, Herr der Mathematik zu werden.

Sie geben dem Computer Zähler und Nenner, der berechnet Ihnen die gesuchten Werte. Wem es dann immer noch zu schwierig ist, der kann sich um diese Zeilen ein Programm basteln, das in der Lage ist, dem Benutzer die ganze Arbeit der Bruchrechnung abzunehmen.

(Jens Budinger/jf/cd)

Programm 4: Code

Ob Geheimagenten, Datenschützer oder heimlich Liebende, alle haben ein existentielles Problem: wie verschlüssel ich meine Meldungen, ohne daß sie jede beliebige Person entziffern kann? Codierung heißt hier das Zauberwort. Für satzgroße Statements hilft Ihnen das Programm "Code" weiter, das Sätze per bewährtem XOR-Verfahren für Unbefugte unkenntlich macht. Das funktioniert so: Sie geben Ihren Satz ein und bestätigen per

RETURN-Taste. Danach wird ein Codewort abgefragt, mit dem der Satz verschlüsselt wird. Nur wer im Besitz dieses Wortes ist, kann die Meldung wieder decodieren.
Die verschlüsselten Daten können auf Diskette gelagert und weitergereicht werden, auf daß der Empfänger des scheinbaren Datensalates mit Hilfe von "Code" den wahren Sinn des Durcheinanders verstehen kann.

(Michael Quast/jf/cd)

Programm 5: Dcode

Wer sich in Assemblerprogrammierung übt, steht bisweilen vor dem Problem, die Anhäufung der entstehenden Daten in DATA-Zeilen eines BASIC-Programms einzubinden. Auch Zeichensätze, die per Symbol-Definition festgelegt wurden, tauchen in langen Datenketten auf, deren Übernahme in BASIC-Programme oft nur mit erheblicher Fleißarbeit verbunden ist.
Dcode nimmt sich dieses Problems an und sorgt für einen präzisen Datentransfer aus einer Binärdatei in ein BASIC-Programm. Die Datei muß auf der Diskette vorhanden sein. Dann kann Dcode gestartet werden. Sie geben dem Programm den Namen der Datei an und den des Basic-Programms, in welches die Daten geschrieben werden sollen. Die erste Zeilennummer wird noch abgefragt, den Rest erledigt das Programm von alleine.
Für Bastler:
– Wenn in Zeile 195 ein CALL &BB06 eingefügt wird, erhält man eine Pause, die für eventuelle Diskettenwechsel genutzt werden kann.
– Eine Veränderung der Variable w in 0 oder 8 bewirkt eine Ausgabe der Datei auf den Bildschirm oder den Drucker.
– Will man nur Teile eines Binärfiles umwandeln, muß man die Variablen b, l und s mit Startadresse, Länge und Einsprungadresse füttern.
– Wer den Memory-Befehl verändert, sollte daran denken, daß der Diskettenbuffer 4 kByte Speicher benötigt.
– Zum Abspeichern eines aktiven Zeichensatzes aus dem RAM gelten folgende Adressen:

```
SYMBOL AFTER x (x=PEEK(&B294) 464/ x=PEEK(&B734) 664/6128)
```

Beginn Zeichensatz: &B296 (464) und &B736 (664/6128)

(Wolf-Thomas Gerner/jf/cd)

Programm 6: Fittometer

Wenn man den lieben langen Tag am Computer sitzt, drohen die wenig strapazierten Muskelansammlungen zu untrainiertem Gewebe zu mutieren. Fettpolster wachsen schneller und man schlappt sichtlich ab. Manche stehen zweimal am Tage auf, um Nahrung zu bekommen, doch reicht diese Form der Bewegung meist nicht aus, um Oberschenkel- und Wadenmuskulatur in solchem Maße zu reizen, daß diese mit dem Wachstum beginnen. Der letzte Ausweg heißt dann Doping. Folge: Immer mehr Menschen wandern vom Computer weg, um Olympiasieger zu werden! Dem muß entgegengewirkt werden.
Das Programm Fittometer dient hier der schnellen Abhilfe dieses Mißstandes, zumindest stärkt es die Armmuskulatur, und bei regelmäßiger Anwendung werden Sie bald zum stärksten Eintipper der Welt.
Laden Sie Fittometer in den CPC und starten Sie das Programm. Sie haben nun die Möglichkeit, den Level zu bestimmen. Zu Beginn empfiehlt sich der Anfänger-Level, denn merke: Auch Arnold S. Arme waren einst spindeldürr, und auch er begann das Training mit Kleinstgewichten.
Dann kann die erste Trainingseinheit starten. Sie rütteln nun den Joystick von links nach rechts und wieder zurück, bis Ihr Arm abfällt, denn die Zeit ist gegen Sie. Durch Ihre Bewegung wird ein Balken auf dem Bildschirm verlängert, und erst wenn dieser eine bestimmte Größe erreicht hat, werden Sie erlöst. Also, ans Werk...

(Thomas Hombert/Frank Puschin/jf/cd)

Programm 7: Fractree

Der Programmname erklärt bereits die Wirkungsweise dieser kleinen Routine. Es werden fraktale Baumstrukturen auf den Bildschirm gezeichnet. Die Vielfalt der verschiedenen Strukturen, die durch Zufallszahlen und Variationen der Parameter möglich sind, ist erstaunlich.
Betrachtet man sich die entstehenden Baumstrukturen genauer, so wird man feststellen, daß ab bestimmten Parametergrenzen die Ähnlichkeit mit in der Natur vorkommenden Bäumen verlorengeht. Diese Frage aber, welche fraktalen Parameter in der Natur vorhanden sind und wieso, ist ein aktuelles Forschungsgebiet, welches in letzter Zeit immer wieder verblüffende Ergebnisse liefert.
Das Programm verlangt die Eingabe vier verschiedener Parameter, wobei s für die Anzahl der Rechenschritte, w für den Öffnungswinkel der Ästepaare, l für die Länge des Stammes und b für die Breite desselben steht.
Experimentieren mit den Ausdrücken für die Variablen b und n liefert beliebig kuriose Baumstrukturen. Mit ein paar Spielereien am Programm läßt sich auch die Rechengeschwindigkeit noch steigern. Wählt man die Winkel größer als 45 Grad, geht die Baumähnlichkeit schnell verloren. Ansonsten heißt es: laden und probieren. Das kurze Programm birgt einige Geheimnisse in sich, die es zu entschlüsseln lohnt.

(Helmut Feuchtgruber/jf/cd)

Programm 8: Hangman

Hangman ist ein kleines Spiel, das so konzipiert ist, daß es langen Spielspaß bringen kann. Die Regeln:
Nach dem Laden müssen Sie Worte eingeben, die keinerlei Beschränkungen unterliegen. Sie haben die Möglichkeit, dem Computer bis zu 999 verschiedene Begriffe einzuspeisen. Beendet wir diese Eingabe mit purem Druck auf die Return-Taste. Dann kann das Spiel beginnen.
Ein noch leerer Galgen erscheint auf dem Bildschirm. Desweiteren macht eine Schablone deutlich, daß es darum geht, Buchstaben zu drücken, um ein Wort zu raten. Die Schablone klärt über die Länge des Wortes auf. Wird ein Buchstabe gedrückt, den das gesuchte Wort enthält, so erscheint er innerhalb der Schablone an den Stellen, an denen er auch im Wort enthalten ist. Kommt er nicht im gesuchten Begriff vor, so wird der Galgen Stück für Stück erweitert. Hat man den Begriff erraten, bevor ein Männchen am Galgen hängt (daher auch der Name des Programms), so gelangt man in die nächste Runde, schafft man es nicht, so ist das Spiel zu Ende.
Auch dieses Produkt lädt den Benutzer zum Basteln ein. Ein paar Vorschläge sollen hier nicht verschwiegen werden.
Wem das Zeichnen des runden Galgenstricks zu lange dauert, der kann dies durch einen kleinen Eingriff in Zeile 11 beheben. Man baut in die Laufanweisung die Schrittlänge 8 oder auch

16 ein (STEP 8 bzw. 16). Der Kreis wird zwar nicht mehr so durchgehend, dafür benötigt der Computer erheblich weniger Rechenzeit.
Wer die Begriffe nicht im voraus wissen will, kann beispielsweise Wortdateien auf Diskette ablagern und mittels kleinem Zusatzprogramm zufällig beliebige Mengen an Worten einladen. Der Fantasie sind hier keine Grenzen gesetzt.

(Fabian Vogelbruch/jf/cd)

Programm 9: Kastler

Geht es ums Essen, so fällt oft der Spruch: Das Auge ißt mit. Der CPC kann zwar eine ganze Menge, aber nach dem Anbieten eines Menüs ist er auch mit seinem Latein am Ende. Wir müssen uns schon auf das Auge beschränken und uns dem Magenwohl anderweitig zuwenden.
Um dem Auge einiges Gutes zu tun, könnten wir beispielsweise das Programm Kastler benutzen. Der Name deutet darauf hin, daß diese wenigen Zeilen sich in irgendeiner Weise mit Kästen beschäftigen.
So können wir mittels Kastler dann auch beliebig große Quadrate oder Rechtecke auf den Bildschirm des CPCs bringen. Nach der Eingabe der Länge und Breite unseres Rechtecks wird noch ein Zerrwert abgefragt, der die Abstände der Rechtecke bestimmt.
Kastler zeichnet daraufhin am linken unteren Bildrand beginnend Kästen in der Größe, die wir durch die eingegebenen Werte bestimmt haben. In kontinuierlicher Folge wird nun gemalt, bis das die rechte obere Bildecke erreicht ist. Was dabei an "äuglichen Genüssen" herauskommt, beurteilen Sie selbst.

(Marco Kunz/jf/cd)

Programm 10: Metronom

Wenn Sie in einem Mietshaus wohnen, bei dessen Fertigstellung der Bauherr besonders an den Steinen zur Errichtung der Wände gespart hat und diese durch etwas stärkere Wellpappe ersetzt hat, dann könnte sich bald ein akustisches Problem einstellen. Es ist durchaus denkbar, daß der Nachbar aus der unteren Etage sich eines Klaviers bereichert, ohne des Spielens auf solch einem monströsen Gerät mächtig zu sein. Sollte dieser sich desweiteren als recht taktlos entpuppen, so werden Sie von da ab mit akustischen Wellen beglückt, über deren ästhetischen Wert man streiten kann.
Zumindest in punkto Takt kann dem guten Mann abgeholfen werden, sofern er einen CPC und das Programm Metronom besitzt.
Metronom gibt in regelmäßigen Abständen ein akustisches und optisches Signal an den Musizierenden weiter, mit dessen Hilfe dieser im Takt bleiben kann. Die Frequenz kann vorab eingestellt werden. Der Benutzer kann stufenlos zwischen 40 und 200 Schlägen pro Minute wählen; da dürfte die Einhaltung der Tempi bestimmter Musikstücke nicht mehr schwerfallen.

(Dominik Vollmer/jf/cd)

Programm 11: Schlange

Keine Angst, obwohl es sich bei diesem Programm um ein paradiesisches Werk handelt, geht es nicht um Verführung nichtsahnender Personen durch irgendwelche Obstsorten. Ziel dieses Spiels ist vielmehr, die zu steuernde Schlange mit möglichst vielen Herzen zu füttern; denn: auch Schlangen sind nur Menschen.
Das Reptil wird per Cursortasten über das Spielfeld gelenkt. Daß sich diese gute Fütterung auf das Wachstum des schleichenden Tieres auswirkt, ist selbstverständlich. Bei der Steuerung ist ebenfalls zu beachten, daß sich die Schlange nicht selber in den Schwanz beißt; denn Schlangen können ja bekanntlich sehr giftig sein!
Am unteren Rand des Bildschirms wird links die Länge der Schlange angezeigt (Score) und rechts die verbleibende Zeit bis zum nächsten Level. Aber jetzt genug beschrieben, schlängeln wir uns lieber ein wenig über den Bildschirm des CPCs.

(O. Raether/jf/cd)

Programm 12: Tabelle

In Heft 5/89 erschien im PC-Teil dieser Zeitschrift das Programm "CONVERT", mit dem man Maße umrechnen kann. Die hier vorliegende Version ist nun für die CPCs geschrieben. Das Programm kennt 12 Maße:
mm – Millimeter
cm – Zentimeter
dm – Dezimeter
m – Meter
km – Kilometer
me – Meile
i – inch (Zoll)
y – Yard
l – Liter
a – Ar
ha – Hektar
f – Fuß
Zuerst müssen Sie das Ausgangsmaß eingeben (z.B. "m" für Meter) und danach das Maß, in das umgerechnet werden soll. Nun verlangt das Programm noch die Potenz, mit der gerechnet werden soll. Geben Sie z.B. 3 ein, so rechnet der Computer in "Kubik" (Kubikdezimeter etc.). Geben Sie jetzt die Größe des Ausgangsmaßes ein. Das Programm rechnet nun die gestellte Aufgabe aus und präsentiert das Ergebnis auf dem Monitor.
Ein Beispiel:
Sie haben eine Fläche von 150 Quadratyards (englisches Maß) und wollen wissen, wieviele Ars das sind. Die Eingabe würde wie folgt lauten:
1. Geben Sie y ein (Yard)
2. Geben Sie a ein (Ar)
3. Geben Sie 2 ein (Potenz 2, weil es hier um das Umrechnen von Flächen geht)
4. Geben Sie 150 ein (es sind ja 150 Yards)
5. Es erscheint auf dem Bildschirm:
150 y↑2 = 1.253094 a
150 Quadratyards entsprechen also einer Fläche von 1.253 Ar.

(Christop Lauble/jf/cd)

Programm 13: Tennis

Welch Balsam auf die Wunden unserer tennisgeplagten Nation durch dieses meisterliche Tennisspiel herabgelassen wird. Alle die, deren Fernseher noch nicht mit dem Kabel verbunden ist, können sich ihr ganz persönliches Wimbledon in einer entzückenden Miniversion auf den CPC bringen. Das Urtennis feiert mit diesem Stück Software Renaissance. Zwei Spieler ha-

ben sich hierbei den durch einen Pixel dargestellten Tennisball zuzuschlagen. Schafft es einer nicht, den Ball, der stetig schneller wird, im Spiel zu halten, so bekommt der Gegner einen Vorteil zugesprochen. Nach dreimaligem Ausgehen verliert der unachtsame Spieler die Partie.
Spieler 1 steuert den Schläger mit den Tasten "Q" und "A", Spieler 2 benutzt die Cursortasten.
Zu Beginn eines Ballwechsels ist Vorsicht angesagt, denn die Schläger erscheinen erst, wenn sie zum ersten Mal gelenkt werden!

(Daniel Adloff/jf/cd)

Programm 14: Wortpuzzle

Jeder, der schreibt, versucht seine Worte möglichst in geordneter Buchstabenfolge zu entwerfen, es sei denn, er legt keinen Wert darauf, daß sein Geschriebenes für andere verständlich ist. Wenn sich ein so ordnungsliebender Satzentwerfer dieses Programm anschaut, so stehen ihm die Haare zu Berge, schüttelt das Spiel Wortpuzzle doch eingegebene Worte derart durcheinander, daß kein Stein mehr auf dem anderen steht. Natürlich ist es nun die Aufgabe des "Aufräumers", die Worte zu richten, um sie wieder leserlich zu gestalten. Die Aktion wird ein Rennen gegen die Zeit. Schafft man es nicht, den ordentlichen Begriff zu rekonstruieren bevor das Stündchen geschlagen hat, so verliert man das Spiel.
Das Rekonstruieren geschieht, indem man jeweils zwei Buchstaben, die nebeneinander stehen, austauscht. Man fährt mittels Cursortasten ein Symbol unter die beiden zu tauschenden Buchstaben und löst den Tauschvorgang mit der COPY-Taste aus. Schnelle Finger und Gedanken sind bei Wortpuzzle gefragt.
Die Länge des Wortes, das der Benutzer zu Spielbeginn einzugeben hat, muß mindestens fünf Buchstaben besitzen. Die Zeit, die man zur Lösung des Problems zur Verfügung hat, steht in Abhängigkeit zur Wortlänge.

Programm 15: Full-Screen-Editor

Das Programm ist ein BASIC-Lader, der eine RSX-Befehlserweiterung einbindet, die einen Full-Screen-Editor ermöglicht. Dies bedeutet, daß man mit dem Cursor beliebig auf dem Bildschirm herumfahren und den Bildschirminhalt ändern kann. Bei dem CPC-Editor ist es ja sonst nur möglich, innerhalb der Eingabezeile den Bildschirminhalt zu editieren.
Mit der Befehlserweiterung ist es so auch möglich, mehrere BASIC-Zeilen zunächst zu verändern, diese anschließend durch mehrmaliges Drücken der ENTER-Taste ins Programm zu übernehmen. Mehrere gleiche oder ähnliche Zeilen können so einfach durch Ändern der Zeilennummern eingegeben werden.
Der Full-Screen-Editor wird durch den RSX-Befehl |Screen aktiviert.
Er ist in der Regel aber nur im Direktmodus sinnvoll, da immer die gesamte Bildschirmzeile als Eingabe angesehen wird, so daß es bei Input-Anweisungen leicht zu Fehlern kommen kann. Bei einer numerischen Eingabe einer Zeile, in der schon Text vorhanden ist, kommt es z.B. zur Fehlermeldung "REDO FROM START".
Da unter BASIC Eingaben 256 Zeichen lang, also über den Bildschirmrand hinausgehend, sein dürfen, muß der Full-Screen-Editor erkennen können, ob zwei oder mehrere Bildschirmzeilen zusammengehören oder nicht.
Um das Programm kurz zu halten, habe ich dieses Problem folgendermaßen gelöst:
Zwei Zeilen gelten als zusammengehörig, wenn eines der beiden letzten Zeichen der oberen Zeile kein Leerzeichen ist; sind die letzten beiden Zeichen einer Zeile Leerzeichen, so ist die Eingabe hier zu Ende. Überflüssige Leerzeichen am Ende einer Eingabezeile werden abgeschnitten.
Diese Methode ist zwar ziemlich simpel, aber auch ziemlich treffsicher. Der Full-Screen-Editor bietet folgende Editiermöglichkeiten:
– Cursor in die linke obere Ecke setzen: Shift + Cursor hoch
– Cursor in die rechte untere Ecke setzen: Shift + Cursor runter
– Cursor auf Zeilenanfang setzen: Ctrl + Cursor links
– Cursor auf Zeilenende setzen: Ctrl + Cursor rechts
– Zeichen Löschen: DEL; Shift + Cursor links
– Leerzeichen einfügen: CLR; Shift + Cursor rechts
Den gewohnten Zeileneditor aktiviert man mit dem RSX-Befehl |Line.
Das Programm läuft nur auf dem 6128!

(S.Ostertun-Gaekel/jf/cd)

Programm 16: Little Shape

Kurzum, das Programm stellt Ihnen zwei neue RSX-Befehle zur Verfügung, die folgendermaßen zu gebrauchen sind:
|READ,x,y,n
holt einen Bildschirmausschnitt, der seine linke obere Ecke an dem Bildschirmpunkt (x/y) hat, 8 Bildschirmbytes breit und 20 Bildschirmzeilen tief ist, in den Speicher ab der Adresse &4000.
|SET,x,y,n
gibt denselben Bildschirmausschnitt wiederum an dem Punkt (x/y) aus. Der Parameter n ist nötig, um mehrere verschiedene Ausschnitte gleichzeitig im Speicher halten zu können – sie sind einfach zu numerieren. n ist also die Nummer des Ausschnittes.
Ein Ausschnitt belegt genau 160 Bytes in Block 1 des CPC-Rams, so daß bis zu 102 solcher "Shapes" verwaltet werden können, was jedoch nicht unbedingt sinnvoll ist, da die Ermittlung der Speicheradresse des jeweiligen Shapes sich bei steigender Anzahl immer mehr in die Länge zieht. Der Parameter n darf sich also von 1 bis 102 bewegen. Wenn sie das Shape n auf Diskette/Kassette sichern wollen, so müssen Sie folgendermaßen vorgehen:

```
Save "Shape.bin",b,&4000+(n-1)*160, 160
```

Geladen wird entsprechend mit:

```
LOAD "shape.bin",&4000+(n-1)*160
```

Jetzt noch etwas zu den Parametern x und y: Das Maschinenprogramm arbeitet mit reellen Punktkoordinaten, das heißt, daß die y-Koordinate, die sich im BASIC ja von 0-398 erstrecken kann, immer durch 2 dividiert werden muß, und daß die x-Koordinate modeorientiert gewählt werden muß:
Mode 0: 0-159
Mode 1: 0-319
Mode 2: 0-639

In Zeile 1025 des Dataladers befindet sich eine kleine Demo. Noch ein Tip zur Definition des Shapes: Zeichnen Sie die Shapes mit einem Grafikprogramm. Hier können Sie sich feste Rahmen definieren, in die die Shapes gezeichnet werden. Wenn man nun die Rahmen an festgelegte Bildschirmpunkte setzt (sie sollten aufgrund der noch erwähnten Laufanweisung regelmäßig gesetzt werden), kann man nach Konstruktion der

Shapes und Sichern des Bildschirms mit einer Laufanweisung die einzelnen Bildschirmausschnitte auslesen und in den Speicher schreiben.

(Carsten Brombach/jf/cd)

Programm 17: 3D-Funktionsplot

Das Sichtbarmachen der abstrakten mathematischen Funktionen ist im Gegensatz zu deren Berechnung oder Erstellung ein äußerst spannendes Thema, sieht man tatsächlich Wirkungsweisen in graphischer Hinsicht, die von dem bloßen Zahlen- und Buchstabenwerk der Funktionen ausgehen.
Mit diesem kleinen Programm kann der CPC uns Funktionsgraphen auf den Monitor bringen und diese noch in die dritte Dimension projizieren, so daß interessanteste Flächen und Gebilde entstehen.
Die Bedienung des Programms ist etwas eigen und bedarf einiger Erläuterungen. Die kleine ENTER-Taste ist mit einer speziellen Funktion belegt, so daß sie nur benutzt werden sollte, wenn das Programm dieses verlangt. Für alle anderen Eingabebestätigungen muß die große ENTER-Taste bzw. RETURN-Taste verwendet werden.
Nach Start des Programms verlangt selbiges die Eingabe einer mathematischen Funktion der Form y=..., wobei y der Funktionswert ist. Als Grundwerte können die Variablen x und z gebraucht werden.
Einige Beispiele:
y=sin(x)*cos(z)*100
y=sin(x)*sin(z)*100
Auch die Funktionen EXP, SQR und TAN können verwendet werden. Es werden vom Programm noch weitere Werte abgefragt, die für die Darstellungsweise nötig sind. Der Neigungswinkel etwa oder die Größe der Quadrate (die Fläche wird aus einzelnen Quadraten gebildet) sowie Werte, die die Dehnung bestimmen.
Experimentieren mit diesem Programm lohnt sich, da es fantastische Bilder auf den Monitor zaubert.

Programme 18 und 19: Kreis und Quadrat

Das Sprichwort sagt: Die Axt im Haus erspart den Zimmermann. Dieses geflügelte Wort auf vorliegendes Programm bezogen würde lauten: Kreis und Quadrat im Haus erspart Hammer und Nagel. Nicht, daß es jemandem gelungen wäre, mit Hilfe des CPCs Nägel in die Wand zu schlagen, um hübsche Bilder daran aufzuhängen, dafür ist der Computer nicht stabil genug, nein, gelungen ist es vielmehr, solche hübschen Bilder zu erstellen, die, sofern ausgedruckt, dann in Rahmen so manche Wand schmücken würden. Es geht um Bilder, die früher mit Hammer, Nagel, Holz und Zwirn angefertigt wurden. Sie erinnern sich: Man nahm ein Stück Holz in Form eines Brettes und malte es in einer zur Farbe des Zwirns harmonisierenden Tönung an. Danach wurden Nägel in geometrischen oder anderen Formen auf das Brett geschlagen. Der Zwirn konnte nun kunstvoll um die Nägel gewunden werden, worauf sich faszinierende Ansichten ergaben.
Die beiden Programme Kreis und Quadrat geben uns die Möglichkeit, solche Gebilde per CPC zu erstellen, und das in einer denkbar einfachen Weise. Das erste Programm "schlägt die Nägel" in runder Form auf den Bildschirm, das zweite in Form eines Quadrates. Durch Eingabe verschiedener Parameter lassen sich Anzahl der Nägel, Abstand zueinander etc. bestimmen, worauf der Computer zu rechnen beginnt und mit der Präsentation toller Grafiken das Programm beendet.

für 464-664-6128

```
1 REM BIG                                        [550]
10 FOR n=&8000 TO &8014:READ a$:POKE n,VAL [2377]
("&"+a$):NEXT
20 DATA fe,02,c0,dd,7e,02,cd,a5,bb,dd,5e,0 [2387]
0,dd,56,01,7d,12,13,7c,12,c9
40 adr%=0:SYMBOL AFTER 0                        [1308]
50 LINE INPUT "String: ",zug$                   [2495]
60 INPUT"x: ",xv                                [827]
70 INPUT "y: ",yv                               [1022]
80 INPUT "Nr.: ",znr                            [1298]
90 IF znr>127 THEN PRINT#8,CHR$(27);CHR$(4 [5891]
9);:znr=znr-128:c=27:d=›1 ELSE PRINT#8,CHR
$(27);"0";:c=0:d=0
100 FOR n=1 TO LEN(zug$)                        [1303]
110 CALL &8000,ASC(MID$(zug$,n,1)),@adr%        [1789]
120 FOR adr=adr% TO adr%+7:a$(adr-adr%)=BI [4909]
N$(PEEK(adr),8):NEXT
130 FOR sp=1 TO 8:FOR x=1 TO xv:PRINT#8:FO [2817]
R r=7 TO 0 STEP-1
140 IF VAL(MID$(a$(r),sp,1))=1 THEN a=c:b= [4325]
d:chr=znr ELSE a=0:b=0:chr=32
150 GOSUB 180                                   [913]
160 NEXT r,x,sp,n                               [783]
170 RUN                                         [243]
180 FOR p=1 TO yv:PRINT#8,CHR$(a);CHR$(b); [2750]
CHR$(chr);CHR$(0);:NEXT
190 RETURN                                      [555]
```

Listing BIG

```
1 REM SCHRIFTVERGROESSERUNG                     [2271]
10 INK 0,0:INK 1,13:PAPER 0:BORDER 6:MODE [1720]
1
20 INPUT "Name des spaeteren Bildes";P$         [4048]
30 PRINT "Inhalt der 1. beiden Zeilen durc [8413]
h Komma getrennt! (max.7 Buchstaben pro Ze
ile)";:INPUT a$,b$
40 IF LEN(a$)>8 OR LEN(b$)>8 THEN SOUND 1, [2778]
239:CLS:GOTO 30
50 a=1:h=250                                    [1310]
60 MODE 2                                       [513]
70 IF a=1 THEN LOCATE 1,25:PRINT a$ ELSE L [2875]
OCATE 1,25:PRINT b$;
80 FOR x=-1 TO 70 STEP 0.3                      [1352]
90 FOR y=1 TO 15                                [1110]
100 IF TEST(x,y) THEN TAG:MOVE 10*x,10*y+h [2460]
:PRINT CHR$(143);
110 NEXT:NEXT                                   [1022]
120 IF a=2 THEN b$=""                           [499]
130 IF b$="" THEN 160 ELSE 140                  [1409]
140 a=2:h=50:LOCATE 1,25:TAGOFF:PRINT SPC( [3803]
10);:GOTO 70
150 LOCATE 1,25:TAGOFF:PRINT SPC(10)            [1698]
160 TAGOFF                                      [1066]
170 LOCATE 1,25:PRINT SPC(20);                  [1474]
180 SAVE P$,B,&C000,&4000                       [1184]
190 CLS                                         [91]
200 TAGOFF:PRINT "Noch ein Bild (J/N)?"         [2907]
210 IF INKEY$="" THEN 210                       [574]
220 IF INKEY(45)<>-1 THEN 10                    [735]
230 IF INKEY(46)<>-1 THEN END                   [1788]
240 GOTO 210                                    [427]
```

Listing Schriftvergrößerung

```
10 REM Calculation                              [1225]
20 MODE 2:CALL &BC02:INK 0,0                    [1896]
30 PRINT"1. Kuerzen (k)   oder   2. Hauptn [8845]
enner berechnen (h) - bitte waehlen Sie
!"
40 a$=UPPER$(INKEY$):IF a$="H" THEN 120 EL [3594]
SE IF a$="K" THEN 50 ELSE 40
50 MODE 1:PRINT"KUERZEN":PRINT:PRINT:INPUT [5092]
 "Zaehler,Nenner ";z,n
60 GOSUB 150:PEN 2:PRINT:PRINT"ggT: ";kle [3488]
70 PEN 3:PRINT:PRINT"gekuerzt:"                 [1980]
80 PRINT:PRINT" ";z/kle                         [976]
90 PRINT" ";STRING$(LEN(STR$(m/kle))-1,"- [2283]
")
100 PRINT" ";n/kle                              [1603]
```

```
110 CALL &BB18:RUN                                    [852]
120 MODE 1:PRINT"HAUPTNENNER BERECHNEN":PR [6704]
INT:PRINT:INPUT "Nenner,Nenner ";z,n
130 GOSUB 150:h=z*n/kle:PEN 2:PRINT:PRINT" [9676]
Hauptnenner:";h:PEN 1:PRINT:PRINT z;:PEN 3
:PRINT"*";h/z:PEN 1:PRINT n;:PEN 3:PRINT"*
";h/n
140 GOTO 110                                          [423]
150 IF z>n THEN gro=z:kle=n ELSE gro=n:kle [2674]
=z
160 m=gro                                             [519]
170 erg=gro/kle:erg=INT(erg)                          [1889]
180 ganz=erg*kle                                      [228]
190 rest=gro-ganz:IF rest=0 THEN RETURN               [2272]
200 gro=kle:kle=rest:GOTO 170                         [963]
```

Listing Calculat

```
1 REM CODE                                            [422]
10 MODE 2:LOCATE 25,1:PRINT" Your security [16541]
 will be saved ! ":LOCATE 30,8:PRINT"Input
 text => 1":LOCATE 30,10:PRINT"Show  text
=> 2":LOCATE 30,12:PRINT"Load  text => 3":
LOCATE 30,14:PRINT"Save  text => 4":LOCATE
 30,16:PRINT"Quit          => 5":a=0
20 WHILE a<1 OR a>5:a=VAL(INKEY$):WEND:ON  [3497]
a GOTO 30,40,50,60,70
30 MODE 2:PRINT"Please input text :":PRINT [10561]
:LINE INPUT a$:PRINT:INPUT"Please input co
de :",c$:CLS:PRINT a$:PRINT:PRINT"Coding .
..":GOSUB 80:GOTO 10
40 CLS:INPUT"Please input code :",c$:GOSUB [9803]
 80:CLS:PRINT"Uncoded text :":PRINT:FOR a=
1 TO LEN(a$):PRINT CHR$(1)MID$(a$,a,1);:NE
XT:GOSUB 80:CALL &BB18:GOTO 10
50 CLS:INPUT"Filename :",n$:OPENIN n$+".co [7514]
d":INPUT#9,anz:a$="":FOR a=1 TO anz:INPUT#
9,c:a$=a$+CHR$(c):NEXT:CLOSEIN:GOTO 10
60 CLS:INPUT"Filename :",n$:OPENOUT n$+".c [8812]
od":anz=LEN(a$):PRINT#9,anz:FOR a=1 TO anz
:PRINT#9,ASC(MID$(a$,a,1)):NEXT:CLOSEOUT:G
OTO 10
70 CLS:PRINT"Bye !":END                               [1367]
80 c=1:FOR a=1 TO LEN(a$):MID$(a$,a,1)=CHR [7251]
$(ASC(MID$(a$,a,1))XOR ASC(MID$(c$,c,1))):
c=c+1+(c=LEN(c$))*LEN(c$):NEXT:RETURN
```

Listing Code

```
1 REM DCODE                                           [1039]
1000 MODE 2:CLEAR:MEMORY &3FFF                        [1487]
1010 DEF FNa=PEEK(adr)+256*PEEK(adr+1)                [1542]
1020 PRINT"DCODE *** Programm zum Umsetzen [13825]
 von Binaerfiles in Basicfiles":PRINT:PRIN
T"Bitte Datendiskette einlegen und eine Ta
ste druecken"
1030 CALL &BB06:CAT                                   [611]
1040 INPUT"Name des Binaerfiles (alt)   :  [3547]
",n$:LOAD n$
1050 INPUT"Name des Basicfiles (neu)    :  [2423]
",nn$
1060 INPUT"Nummer der 1.Zeile           :  [923]
",z
1070 adr=&A76A:b=FNa-1                                [1488]
1080 adr=&A76D:l=FNa                                  [1278]
1090 adr=&A76F:IF FNa THEN s=FNa ELSE s=b+ [2235]
1
1100 w=9:n=8                                          [397]
1110 OPENOUT nn$                                      [559]
1120 PRINT#w,z"MEMORY &"HEX$(b,4)                     [2437]
1130 z=z+10:PRINT#w,z"FOR i=1 TO"l                    [1985]
1140 z=z+10:PRINT#w,z"READ a$:POKE HIMEM+i [5183]
,val("CHR$(34)"&"CHR$(34)"+a$)"
1150 z=z+10:PRINT#w,z"NEXT: REM Einsprung: [3733]
&"HEX$(s,4)
1160 p=0                                              [257]
1170 IF p=l THEN 1220                                 [629]
1180 IF p MOD n=0 THEN z=z+10:PRINT#w:PRIN [2655]
T#w,z"DATA "; ELSE PRINT#w,",";
1190 p=p+1                                            [78]
1200 PRINT#w,HEX$(PEEK(b+p),2);                       [1796]
1210 GOTO 1170                                        [305]
1220 PRINT#w:CLOSEOUT                                 [1676]
```

Listing Dcode

```
1 REM FITTOMETER                                      [1828]
10 MODE 1:PRINT"1) PROFI":PRINT"2) AMATEUR [8171]
":PRINT"3) ANFAENGER":PRINT:INPUT a:IF a=1
 THEN h=0.45 ELSE IF a=2 THEN h=0.35 ELSE
h=0.08
20 m=2:p=250:MODE 1:z=TIME:LOCATE 12,1:PRI [3076]
NT"FITTO METER"
30 FOR I=1 TO 4:MOVE 306,100+i*50:DRAWR 10 [9937]
,0:MOVE 310,100:DRAWR 0,200:NEXT I:MOVE 17
0,255:TAG:PRINT"BAD";:MOVE 170,205:PRINT"O
KAY";:MOVE 170,153:PRINT"GREAT";:TAGOFF
40 PLOT 320,300:DRAW 320,p,0                          [1881]
50 IF INKEY(74)=0 AND m=2 THEN p=p-1:m=1              [2055]
60 IF INKEY(75)=0 AND m=1 THEN p=p-1:m=2              [1062]
70 p=p+h:LOCATE 1,1:PRINT INT(s):s=(TIME-z [3471]
)/300
80 CALL &BD19:PLOT 320,300:DRAW 320,p,1:PL [2579]
OT 320,300:DRAW 320,p,1
90 IF INT(s)=46 THEN PRINT:PRINT"Die 45 Se [4560]
kunden sind schon um..!":GOTO 130
100 IF p>300 THEN PRINT:PRINT"Schon nach " [6260]
;INT(s);"Sekunden kaputt ?":GOTO 130
110 IF p<100 THEN PRINT:PRINT"Spitze ! Nur [6891]
 ";INT(s);"Sekunden gebraucht":CLEAR INPUT
:h=h+0.1:FOR zeit=1 TO 1500:NEXT:GOTO 20
120 GOTO 40                                           [378]
130 CLEAR:CLEAR INPUT:PRINT:INPUT"Nochmal  [3957]
";no$
140 IF UPPER$(no$)="J" THEN RUN ELSE END              [2510]
```

Listing Fittometer

```
1 REM FRACTREE                                        [890]
10 MODE 2:DEG:c=2048:DIM w(c),x(c),y(c):w( [3889]
1)=90:q=1:j=c
20 INPUT "S:";m:INPUT "W:";g:INPUT "L:";b: [4198]
INPUT "B:";a:CLS:GOSUB 100
30 FOR i=1 TO m-1                                     [448]
40 a=a/COS(g)/2:q=q*2:j=c/q                           [1304]
50 FOR k=1 TO c STEP j:r=k-j                          [1980]
60 IF (k+1)/j MOD 4=0 THEN d=b ELSE d=b*n             [2086]
70 IF (k-1)/j MOD 2=0 THEN x(k)=x(k)+d*COS [8557]
(w(k)):y(k)=y(k)+d*SIN(w(k)):w(k)=w(k)+g:E
LSE x(k)=x(r)+SIN(w(r))*a:y(k)=y(r)-COS(w(
r))*a:w(k)=w(r)-2*g
80 NEXT                                               [350]
90 b=b*RND(1)+20:n=RND(2)+0.4:GOSUB 100:NE [2771]
XT:END
100 FOR k=1 TO c STEP j                               [1648]
110 x=COS(w(k)):y=SIN(w(k)):w=b*y:z=b*x               [2612]
120 IF (k+1)/j MOD 2<>0 THEN w=w*n:z=z*n              [988]
130 u=a*y:v=a*x:MOVE x(k)+300,y(k):MOVER u [4667]
,-v:DRAWR z,w:MOVER-u,v:DRAWR-z,-w
140 NEXT:RETURN                                       [940]
```

Listing Fractree

```
1 REM HANGMAN                                         [684]
10 DIM w$(999):RANDOMIZE TIME                         [688]
20 INPUT"Wort";w$(j):IF w$(j)=""THEN j=j-1 [3984]
 ELSE j=j+1:GOTO 20
30 DIM a$(30):o$=w$(INT(RND*j)):a=LEN(o$): [10481]
MODE 1:DRAW 638,0:DRAWR 0,398:DRAWR -638,0
:DRAWR 0,-398:MOVE 550,80:DRAW 550,250:DRA
W 450,250:MOVE 500,250:DRAW 550,200:LOCATE
 2,3:PRINT STRING$(a,"-")
40 LOCATE 2,5:INPUT"Buchstabe";b$:FOR i=0  [4040]
TO t:IF a$(i)=b$ THEN 40
50 NEXT:FOR i=1 TO a:IF MID$(o$,i,1)=b$ TH [4723]
EN LOCATE i+1,3:PRINT b$:d=1:p=p+1
60 NEXT                                               [350]
70 IF d=0 THEN 80 ELSE IF p=a THEN 100 ELS [2854]
E d=0:a$(t)=b$:t=t+1:GOTO 40
80 d=1:f=f+1:ON f GOTO 90,110,120,130,140, [3439]
150,160
90 MOVE 470,250:DRAW 470,220:GOTO 70                  [1998]
100 LOCATE 16,23:PRINT"GEWONNEN":GOTO 170             [2129]
110 x=470:y=200:DEG:FOR c=270 TO 630:PLOT  [5260]
x+20*COS(c),y+20*SIN(c):NEXT:GOTO 70
120 DRAW x,120:GOTO 70                                [683]
130 MOVE x,165:DRAW 450,145:GOTO 70                   [1445]
140 MOVE x,165:DRAW 490,145:GOTO 70                   [1307]
150 MOVE x,120:DRAW 450,99:GOTO 70                    [1610]
160 MOVE x,120:DRAW 490,99:LOCATE 16,23:PR [2872]
INT"VERLOREN"
170 ERASE a$:t=0:p=0:d=0:f=0:CALL &BB18:GO [2011]
TO 30
```

Listing Hangman

```
10 REM KASTLER                                        [865]
20 'written by Marco Kunz                             [1857]
25 INK 0,0:INK 1,26:BORDER 0:GRAPHICS PEN  [4238]
1:PEN 1:MODE 2:CLS:x1=100:x2=540:y1=50:y2=
350
27 MOVE x1,y1:DRAW x1,y2:DRAW x2,y2:DRAW x [5195]
2,y1:DRAW x1,y1:IF y1<10 THEN 30 ELSE y1=y
1-3:y2=y2+3:x1=x1-3:x2=x2+3:GOTO 27
```

```
30 LOCATE 18,6:PRINT"KASTLER V1.0  written [5047]
 by Marco Kunz / 1 Kbyte"
40 LOCATE 34,9:INPUT"X-Staerke ";xs:LOCATE [12876]
 34,12:INPUT"Y-Staerke ";ys:LOCATE 34,15:I
NPUT"ZERR-Wert ";z:LOCATE 33,18:PRINT"Tast
e zum Beginn":CALL &BB18:GRAPHICS PEN 1:CL
S:x=20:y=20
50 x=x+z+(z/2):y=y+z:FRAME:PLOT x+xs,y+ys: [7411]
DRAW x+xs,y-ys:DRAW x-xs,y-ys:DRAW x-xs,y+
ys:DRAW x+xs,y+ys:IF y>360 THEN 60 ELSE 50
60 CALL &BB18:RUN                          [852]
```
Listing Kastler

```
10 REM METRONOM written by Dominik Vollmer [2522]
20 MODE 1:BORDER 0:INK 0,5:INK 1,26:INK 2, [1905]
10
30 WINDOW 3,38,3,23:PAPER 2:CLS:f=1        [2139]
40 LOCATE 15,2:PEN 1:PRINT"METRONOM";SPC(2 [3644]
7);STRING$(10,154)
50 LOCATE 1,21:PRINT CHR$(24);SPC(5);" 198 [3352]
9 by Dominik Vollmer";
60 PRINT SPC(6);CHR$(24);                  [1127]
70 LOCATE 2,9:INPUT"Wieviele Schlaege in d [3871]
er Min.";a
80 LOCATE 2,9:PRINT SPC(35):IF a<40 OR a>2 [2866]
00 THEN 70
90 WINDOW 5,36,7,20:PAPER 2:CLS:PLOT 260,1 [2920]
00,1:DRAW 380,100:DRAW 330,240
100 DRAW 310,240:DRAW 260,100:PLOT 275,140 [2356]
:DRAW 365,140
110 TAG:MOVE 250,290:PRINT"< SPACE >";:MOV [4357]
E 282,125:PRINT a;:TAGOFF
120 PLOT 320,140,1:DRAW 370,230            [1318]
130 FOR w=1 TO 17950/a:IF INKEY(47)<>-1 TH [3115]
EN 190 ELSE NEXT w:SOUND 1,142,4,15
140 BORDER 26:FOR x=1 TO 20:NEXT:BORDER 0: [3773]
IF f=0 THEN 170
150 PLOT 320,140,2:DRAW 370,230:PLOT 320,1 [2822]
40,1
160 DRAW 270,230:PLOT 380,100,1:DRAW 330,2 [2801]
40:f=0:GOTO 130
170 PLOT 320,140,2:DRAW 270,230:PLOT 320,1 [1992]
40,1
180 DRAW 370,230:PLOT 310,240,1:DRAW 260,1 [3453]
00:f=1:GOTO 130
190 i$=INKEY$:IF i$<>""THEN 190            [1909]
200 CLS:WINDOW 3,38,3,23:f=1:GOTO 70       [1080]
```
Listing Metronom

```
10 REM SCHLANGE                            [696]
20 '1988 O.Raether                         [831]
30 DIM x(1000),y(1000)                     [725]
40 l=3                                     [186]
50 MODE 1:PRINT STRING$(40,CHR$(207));     [2609]
60 FOR a=2 TO 23:PRINT CHR$(207)SPACE$(38) [3334]
CHR$(207);:NEXT
70 PRINT STRING$(40,CHR$(207))             [1172]
80 FOR a=1 TO 10:GOSUB 220:NEXT            [2122]
90 k=l:r=4:FOR a=1 TO k:x(a)=10:y(a)=10:NE [2416]
XT
100 a$=INKEY$+" "                          [482]
110 IF ASC(a$)>239 AND ASC(a$)<244 THEN r= [1775]
ASC(a$)-239
120 k=k+1:IF k>1000 THEN 50                [1510]
130 y(k)=y(k-1):x(k)=x(k-1)                [1315]
140 IF r=1 THEN y(k)=y(k)-1 ELSE IF r=2 TH [8749]
EN y(k)=y(k)+1 ELSE IF r=3 THEN x(k)=x(k)-
1 ELSE x(k)=x(k)+1
150 c=TEST(x(k)*16-8,414-y(k)*16)          [2465]
160 IF c=1 THEN END ELSE IF c=2 THEN l=l+2 [2110]
:GOSUB 220
170 LOCATE x(k-1),y(k-1):PRINT CHR$(202);  [1598]
180 LOCATE x(k),y(k):PRINT CHR$(239+r);    [1984]
190 LOCATE x(k-l),y(k-l):PRINT CHR$(32);   [3494]
200 LOCATE 11,25:PRINT l,1000-k;:FOR a=1 T [2258]
O 100-k/10:NEXT
210 GOTO 100                               [417]
220 PEN 2:LOCATE RND*37+2,RND*20+2:PRINT C [3742]
HR$(228);:PEN 1:RETURN
```
Listing Schlange

```
10 '*******************************        [1223]
20 '* Umrechnungstabelle fuer CPC *        [1443]
30 '*                             *        [175]
40 '*     written by C.Lauble     *        [1533]
50 '*                             *        [175]
60 '*         in 5/1989           *        [662]
70 '*******************************        [1223]
80 '                                       [117]
90 '                                       [117]
100 DATA mm,1,cm,10,dm,100,m,1000,km,10000 [4933]
00,me,1609300,y,914,f,304.8,1,100,a,10000,
ha,100000,i,25.4
110 INK 0,0:INK 1,26:PAPER 0:PEN 1:MODE 2: [2378]
BORDER 0
120 LOCATE 15,2:PRINT"U  m  r  e  c  h  n [4460]
  u  n  g  s  t  a  b  e  l  l  e"
130 LOCATE 14,3:PRINT STRING$(54,208)      [1732]
140 LOCATE 31,4:PRINT"written by C. Lauble [3414]
"
150 LOCATE 1,6:PRINT"Das hier vorliegende  [6840]
Programm wurde bereits im Locomotive BASIC
 V1.21 von"
160 PRINT"Sven Kaestner programmiert. Dies [7338]
e Version ist nun auf den CPC's lauffaehig
 !!!"
170 LOCATE 1,10:PRINT"Geben Sie das Mass e [7108]
in, von dem aus gerechnet werden soll :"
180 LINE INPUT m1$                         [1095]
190 LOCATE 1,13:PRINT"Geben Sie das Mass e [5732]
in, in das gerechnet werden soll :"
200 LINE INPUT m2$                         [1093]
210 LOCATE 1,16:INPUT "In welcher Potenz s [5412]
oll gerechnet werden (1-3) ";p%
220 IF p%>0 AND p%<4 THEN 240              [1876]
230 PRINT CHR$(7):RESTORE:GOTO 10          [2436]
240 FOR x%=1 TO 12:READ m$,m1:IF m1$<>m$ T [3985]
HEN NEXT x%:GOTO 230
250 RESTORE                                [621]
260 FOR x%=1 TO 12:READ m$,m2:IF m2$<>m$ T [3451]
HEN NEXT x%:GOTO 230
270 PRINT "Wieviele ";m1$;:INPUT a         [2439]
280 ma1=LOG10(m1):ma2=LOG10(m2)            [1859]
290 d=ABS(ma1-ma2)                         [886]
300 e1=10^(p%*d)                           [933]
310 IF ma1>ma2 THEN e=a*e1 ELSE e=a/e1     [1925]
320 PRINT"Ergebnis :";a;m1$;               [1080]
330 IF p%=2 AND m1$<>"a" AND m1$<>"ha" THE [2340]
N PRINT"^2";
340 IF p%=3 AND m1$<>"l" THEN PRINT"^3";   [2684]
350 PRINT"  = ";e;m$;                      [1407]
360 IF p%=2 AND m2$<>"a" AND m2$<>"ha" THE [2695]
N PRINT"^2";
370 IF p%=3 AND m2$<>"l" THEN PRINT"^3";   [1980]
380 PRINT:PRINT:PRINT"Weitere Berechnung d [4051]
urchfuehren (J/N) ?"
390 a$=INKEY$:IF a$="j" OR a$="J" THEN RUN [1806]
400 IF a$<>"n" AND a$<>"N" THEN 390        [1146]
```
Listing Tabelle

```
1 REM MINI-TENNIS                          [1032]
10 MODE 1:PEN 1:LOCATE 16,5:PRINT"MINI-TEN [8236]
NIS":PLOT 270,200,3:DRAWR 100,0:PLOT 270,1
50:DRAWR 100,0:CALL &BB06
20 z=1:y=175:a=175:r=320:s=175:u=1:t=z:PLO [7527]
T 270,150,0:DRAWR 0,50:PLOT 370,150:DRAWR
0,50
30 IF INKEY(67)=0 AND y<194 THEN y=y+2 ELS [4391]
E IF INKEY(69)=0 AND y>152 THEN y=y-2 ELSE
 50
40 PLOT 270,y-2,0:DRAWR 0,10:PLOT 270,y,1: [2263]
DRAWR 0,6
50 IF INKEY(0)=0 AND a<194 THEN a=a+2 ELSE [2831]
 IF INKEY(2)=0 AND a>152 THEN a=a-2 ELSE 7
0
60 PLOT 370,a-2,0:DRAWR 0,10:PLOT 370,a,1: [2539]
DRAWR 0,6
70 PLOT r,s,0:r=r+t:s=s+u:IF r<260 THEN i= [5249]
2:GOTO 110 ELSE IF r>380 THEN i=1:GOTO 110
80 IF TEST(r,s)>0 THEN 90 ELSE PLOT r,s,2: [2531]
GOTO 30
90 IF r>368 THEN t=-z:z=z*1.2 ELSE IF r<27 [8274]
2 THEN t=z:z=z*1.2 ELSE IF s>190 THEN u=-1
 ELSE IF s<160 THEN u=1
100 r=r+t:s=s+u:GOTO 30                    [1475]
110 p(i)=p(i)+1:IF p(i)=3 THEN LOCATE 9,20 [7600]
:PRINT "PLAYER "i" HAT GEWONNEN !":FOR i=1
 TO 300:SOUND 1,RND*500,1:NEXT:RUN ELSE 20
```
Listing Tennis

```
1 REM WORTPUZZEL                           [1814]
10 h=150:h$="Merlyn":INK 0,0:INK 1,26:INK  [5108]
2,7,21:INK 3,17:BORDER 0
20 MODE 1:PEN 1:INPUT"Bitte Wort eingeben! [5327]
 ",w$:l=LEN(w$):IF l<5 OR l>38 THEN 20
30 t=3*l:w$=UPPER$(w$):n$=w$:LOCATE 5,5:PR [6032]
INT"Highscore"h"by "h$
```

```
40 FOR n=1 TO 4*l:x=INT(RND*(l-1))+1:GOSUB [3452]
 160:NEXT
50 LOCATE 15,17:PEN 2:PRINT"TASTE=Start"   [2848]
60 CALL &BB18:x=1:EVERY 25 GOSUB 180        [1756]
70 WHILE t:PEN 3:IF n$=w$ THEN 120          [1526]
80 LOCATE 15,17:PRINT"  Zeit:"t:LOCATE x+( [4657]
38-l)\2,11:PRINT" "STRING$(2,240)" "
90 IF NOT INKEY(8)AND x>1 THEN x=x-1 ELSE  [2940]
IF NOT INKEY(1)AND x<l-1 THEN x=x+1
100 IF NOT INKEY(9)THEN GOSUB 160           [949]
110 WEND:PEN 2:LOCATE 15,17:PRINT"Zeit ist [3102]
 aus!"
120 d=REMAIN(0):s=ROUND(50*t/SQR(l)):PEN 1 [4490]
:LOCATE 15,19
130 PRINT CHR$(7)"Punkte:"s:WHILE INKEY$<> [3874]
"":WEND
140 IF s>h THEN h=s:LOCATE 6,21:PEN 2:INPU [2440]
T"Highscore! Name: ",h$
150 LOCATE 15,24:PEN 3:PRINT"TASTE=Neustar [3410]
t":CALL &BB18:GOTO 20
160 MID$(n$,x)=MID$(n$,x+1,1)+MID$(n$,x,1) [1867]
170 LOCATE(42-l)\2,10:PEN 1:PRINT n$:SOUND [2284]
 1,300,1,15:RETURN
180 SOUND 2,1000,2,10:t=t-1:RETURN          [3109]
```

Listing Wortpuzzle

```
1 REM SCREEN-EDITOR                         [791]
10 r=&A400:MEMORY r-1:FOR i=0 TO 2:READ c$ [6621]
:FOR j=1 TO 204:a=ASC(MID$(c$,j))-48:IF a>
12 THEN a=a-5:IF a>43 THEN a=a-6
20 FOR k=1 TO 6:b$=b$+MID$(BIN$(a,6),k,1): [6603]
IF LEN(b$)=8 THEN POKE r,VAL("&x"+b$):r=r+
1:b$=""
30 NEXT k,j,i:CALL &A400                    [701]
100 DATA 8QmZ0EaYkx6w4?H38eEM2pB3IYJ5nY19H [14446]
gIODgwV0ekM1Hv38GuYAZux8ZyxmLvBKfgXhm3snM6
vnEOvnMGvzbwk8zsUp5evACny42V0zUoUtApMdHsBn
Jev8MegTfQy2g1OinT9zj0k2Ttz?1vBKfgMkxPkzUk
kjMQadAOOPyYM5VUM61cM71UW62YMP1Uw6?EMcVUX6
2gy
110 DATA 21U?DUcM1Xs8608y2QpOiwAqd2410Qppi [16029]
wAqd2sN60rBS@ga0Qppi1XDnLWvnIqZN1XmnLWvtQp
MdGf?dQppim6?f7vBKfjXuzLBS@j1hypGuyDpDW3AK
fj18vQUtS7BRPgMZQpsirmpvU4cxSLBK?IemeLBRPg
VWe1ynJevsyjpnLWvwPThIijXxHs8813BKfj1xPQc2
2By
120 DATA nJevwLvpwGCr8BMy8ApOiy7BRPgy2ApOi [16063]
1WitRLBOPhuZ3nFsQbBS@ga0G@?dSLBRPjV@GUSDUX
BKfjBM@jy823bDUXBKfjBM@jy823P?geZnLKv4Meg3
UOM1Hs9nJevnIqZFwpUi18HS3oy2QFOikoc4V3j61c
HzW0Ur1gP6VAHzW0UpewGumhyzW30DUXBKfgq00kUw
QY0
```

Listing Full-Screen-Editor

```
1000 ' ### LITTLE SHAPE VON CARSTEN BROMBA [1954]
CH 1989 ###
1010 MEMORY &9FFF:FOR i=&A000 TO &A094:REA [3726]
D byte$:POKE i,VAL("&"+byte$):NEXT
1020 CALL &A000                             [637]
1025 MODE 0:PEN 1:PRINT:PRINT"#";:PEN 15:P [7714]
RINT"#":¦READ,0,199,1:FOR y=199 TO 25 STEP
 -4:¦SET,100,y,1:NEXT:CLS:GOTO 1025:END
1030 DATA 01,13,a0,21,1c,a0,cd,d1          [1422]
1040 DATA bc,01,20,a0,21,28,a0,cd          [1138]
1050 DATA d1,bc,c9,17,a0,18,15,52          [2024]
1060 DATA 45,41,c4,00,00,00,00,00          [1405]
1070 DATA 24,a0,18,2b,53,45,d4,00          [1266]
1080 DATA 00,00,00,00,fe,03,c0,cd          [1822]
1090 DATA 75,a0,cd,88,a0,06,14,c5          [1665]
1100 DATA d5,e5,cd,1d,bc,06,08,7e          [1682]
1110 DATA fd,77,00,23,fd,23,10,f7          [1502]
1120 DATA e1,d1,c1,2b,10,e9,c9,fe          [1229]
1130 DATA 03,c0,cd,75,a0,cd,88,a0          [1336]
1140 DATA cd,19,bd,06,14,c5,d5,e5          [1188]
1150 DATA cd,1d,bc,06,08,fd,7e,00          [906]
1160 DATA 77,23,fd,23,10,f7,e1,d1          [1254]
1170 DATA c1,2b,10,e9,c9,dd,7e,00          [1796]
1180 DATA fd,21,00,40,47,c5,06,a0          [1339]
1190 DATA fd,23,10,fc,c1,10,f6,c9          [1766]
1200 DATA dd,6e,02,dd,66,03,dd,5e          [1062]
1210 DATA 04,dd,56,05,c9                   [979]
```

Listing Little Shape

```
1 REM 3D-FUNKTIONSPLOTTER                   [1035]
10 MODE 2:DEG:ORIGIN 320,200                [1222]
20 PRINT "Geben Sie die Funktion ein, dana [5967]
ch die kleine ENTER-Taste !"
30 KEY 11,CHR$(248)+"270 "+CHR$(13)+"GOTO  [3467]
50"+CHR$(13)
40 END                                      [110]
50 INPUT "Winkel ",w                        [1053]
60 INPUT "Quadratgroesse ",gr               [1866]
70 INPUT "X-Dehnung ",dx                    [1067]
80 INPUT "Y ",dy                            [598]
85 CLS                                      [91]
90 zx=COS(w):zy=SIN(w)                      [1873]
100 FOR z=180 TO -180+gr STEP -gr           [1865]
110 FOR x=-180 TO 180-gr STEP gr            [2531]
120 GOSUB 270                               [927]
130 bx1=dx*(x+z*zx):by1=dy*(y+z*zy)         [1323]
140 x=x+gr                                  [1292]
150 GOSUB 270                               [927]
160 bx2=dx*(x+z*zx):by2=dy*(y+z*zy)         [3193]
170 z=z-gr                                  [751]
180 GOSUB 270                               [927]
190 bx3=dx*(x+z*zx):by3=dy*(y+z*zy)         [2790]
200 x=x-gr                                  [1236]
210 GOSUB 270                               [927]
220 bx4=dx*(x+z*zx):by4=dy*(y+z*zy)         [834]
230 z=z+gr                                  [839]
240 MOVE bx1,by1:DRAW bx2,by2:DRAW bx3,by3 [5122]
:DRAW bx4,by4:DRAW bx1,by1
250 NEXT x,z:END                            [1038]
260 '                                       [117]
270 '                                       [117]
280 RETURN                                  [555]
```

Listing 3D-Funktionsplot

```
1 REM KREIS                                 [625]
10 MODE 1                                   [506]
20 INPUT"Anzahl der Punkte ";p              [1979]
30 IF p/2<>INT(p/2) THEN 10                 [1019]
40 INPUT"Versatz";v                         [1432]
50 INPUT"Radius x-Richt.";rx                [2380]
60 INPUT"Radius y-Richt.";ry                [1912]
70 INPUT"Drall";d                           [1081]
80 IF v<>INT(v) OR v<0 OR v>(p-2)/2 THEN 1 [2136]
0
90 CLS:DIM f(2*p,2):xm=320:ym=200           [1006]
100 FOR i=1 TO p                            [406]
110 f(i,1)=xm+rx*SIN(2*PI*(i-1)/p)          [2690]
120 f(i,2)=ym+ry*COS(2*PI*(i-1)/p)          [1186]
130 PLOT f(i,1),f(i,2)                      [914]
140 NEXT i                                  [375]
150 p1=p/2+v                                [453]
160 FOR i=1 TO p                            [406]
170 xa=f(i,1):ya=f(i,2)                     [1669]
180 p1=p1+d                                 [117]
190 IF p1>p THEN  p1=p1-p                   [1012]
200 xe=f(p1,1):ye=f(p1,2)                   [1475]
210 MOVE xa,ya:DRAW xe,ye                   [1584]
220 NEXT                                    [350]
```

Listing Kreis

```
1 REM QUADRAT                               [401]
10 MODE 2                                   [513]
20 INPUT "Anzahl Punkte:";p                 [1365]
30 IF p/4 <> INT(p/4)  THEN 10              [1027]
40 INPUT"Abstand Quadrat:";ab               [1568]
50 INPUT"Versatz:";v                        [1822]
60 IF v<>INT(v) OR v<0  OR v>p/2 THEN 50    [1807]
70 a=p/4:e=(2*ab)/a                         [1398]
80 DIM f(2*p,2)                             [694]
90 ORIGIN 320,200                           [215]
100 FOR i=1 TO p                            [406]
110 IF j=a THEN j=0                         [719]
120 IF i<=a THEN f(i,1)=-ab+e*j:f(i,2)=ab  [2345]
130 IF i<=2*a AND i>a THEN f(i,1)=ab:f(i,2 [1356]
)=ab-e*j
140 IF i<=3*a AND i>2*a THEN f(i,1)=ab-e*j [1630]
:f(i,2)=-ab
150 IF i<=4*a AND i>3*a THEN f(i,1)=-ab:f( [3136]
i,2)=-ab+e*j
160 j=j+1:NEXT:CLS                          [709]
170 FOR i=1 TO p:PLOT f(i,1),f(i,2):NEXT   [1143]
180 'k=2*a+v                                [449]
190 FOR i=1 TO p                            [406]
200 xa=f(i,1):ya=f(i,2)                     [1669]
210 k=k+v:IF k>p THEN k=k-p                 [1860]
220 xe=f(k,1):ye=f(k,2)                     [1769]
230 MOVE xa,ya:DRAW xe,ye                   [1584]
240 NEXT                                    [350]
```

Listing Quadrat

Neues vom Bankmanager

Eine Verbesserung

Wer einen CPC 6128 käuflich oder anders erworben hat, erhält gleichzeitig zwei Systemdisketten. Auf Seite 1 der Systemdiskette ist ein Programm mit dem Namen "BANKMAN.BAS" vorhanden. Dieses lädt nach Start "BANKMAN.BIN" nach und ermöglicht uns die Nutzung der zweiten 64 kByte des 6128 unter BASIC. Das Handbuch erklärt auch die nähere Umgangsweise damit. Bei den dort gegebenen Beispielen läuft die ganze Sache auch reibungslos ab. Wenn man nun aber beginnt, eigene Programme unter Zuhilfenahme der RSX-Erweiterung zu erstellen, kann es leider sehr leicht zu Programmabstürzen, Fehlermeldungen oder unkorrekter Verarbeitung der übergebenen Variablen kommen. Und das "vergällt" einem sehr schnell die Freude an der Nutzung der zweiten RAM-Bank. Dieser Artikel soll Ihnen helfen, bei der Benutzung der zweiten RAM-Bank ohne großen Ärger auszukommen.

Das Handbuch des CPC 6128 erklärt die Benutzung des BANKMANAGERS leider nicht abschließend und ausführlich genug, um ohne weiterführende Kenntnisse eine reibungslose Arbeit damit zu gewährleisten.

Bankmanager als Bildspeicher

Die Arbeit mit dem Programm zur schnellen Bearbeitung und Austausch von Bildschirminhalten wird im Handbuch ausreichend erklärt. Anzumerken ist hier nur, daß die Bild- und Datenverwaltung nicht gleichzeitig benutzt werden sollte, wenn Sie im Umgang mit dem Bankmanager noch nicht richtig vertraut sind. Denn was nützt Ihnen die schönste Grafik, wenn diese im zweiten Block der zweiten 64 kByte abgelegt ist und durch Einschreiben von Stringvariablen zerstört wird. Des weiteren ist darauf zu achten, in welchem Mode die Bilder bearbeitet werden. Am besten jedesmal einer numerischen Variablen den aktuellen Mode mit übergeben. Wird das nicht beachtet, kann es zu unliebsamen Überraschungen kommen, wenn Bilder in verschiedenen Modes getauscht werden. Am Ende des Artikels ist ein Berechnungsbeispiel aufgeführt, welches aber nur mit weiteren Kenntnissen aus diesem Text verstanden werden kann, aber für die gleichzeitige Bearbeitung von Grafiken und Daten mit dem Bankmanager wichtig sind (siehe auch Handbuch des CPC 6128, Kap. 8, Seite 2 bis 4).

Bankmanager als Datenverwaltung

Das Programm verarbeitet ausschließlich Daten, welche vorher einer Stringvariablen übergeben wurden. Zwar werden numerische Werte ohne Fehlermeldung oder Fehlercode (darauf kommen wir noch später zu sprechen) angenommen, aber nicht verarbeitet. Folglich müssen Daten immer mit STR$ oder DEC$ in Stringvariable umgewandelt werden, um diese dann in den zweiten 64 kByte ablegen zu können. Abgesehen davon sollte von einer vorher festgelegten Länge eines Datensatzes nicht abgewichen werden (ähnlich wie bei der Behandlung von Variablen in der relativen Datenverwaltung mit Diskettenbetrieb).

Nun geht es ans Eingemachte

Zu Beginn müssen Sie vor aller Überlegung den Bankmanager mit RUN"BANKMAN.BAS" starten, um die neuen RSX-Befehle zur Verfügung zu haben. Vor jeder neuen Inbetriebnahme eines Programmes, welches mit dem erweiterten Befehlssatz arbeitet, muß die RSX-Erweiterung gestartet werden. Das kann auf die Dauer mit den ganzen Diskettenwechseln lästig werden. Leichter ist es da, die benötigten Programme von der Systemdiskette auf die jeweilige Arbeitsdiskette abzuspeichern. Das BASIC-Listing des Bankmanagers sträubt sich allerdings schon gegen eine normale Überspielung. Kein Wunder, das Programm wurde mit dem Zusatz "P" abgespeichert, also LIST-geschützt. Entweder Sie heben den Listschutz mit einem Directory-Editor-Programm auf, oder Sie kopieren beide Programme unter CP/M+ mit PIP.

Vorgehensweise: Seite 1 der Systemdisketten einlegen und |CPM gefolgt von PIP eingeben. Dann "B:=A:Bankman.Bas" tippen (gilt nicht für Besitzer eines zweiten Laufwerkes) und den Anweisungen auf dem Bildschirm folgen. Das gleiche mit "Bankman.Bin" wiederholen. Nun haben Sie den Bankmanager vollständig auf Ihrer Arbeitsdiskette stehen. Dennoch muß nun immer noch vor Start eines Programms RUN"BANKMAN.BAS" eingegeben werden. Für alle diejenigen, die den binären Teil etwas professioneller innerhalb Ihres Programms laden wollen, können die Zeilen aus Listing 1 (BANKMAN1.BAS) an den Beginn Ihres Listings setzen.

Als darauffolgende Zeile empfehlen wir, gleich mit |BANKOPEN,(x) festzustellen, ob tatsächlich der Bankmanager vorhanden ist. Denn wird hier die Fehlermeldung UNKNOWN COMMAND ausgegeben, kann in einer Fehlerabfangroutine entsprechend darauf reagiert werden. Vor Neustart des Programms dann allerdings den Computer mit CTRL+SHIFT+ESC zurücksetzen. Nun können Sie "BANKMAN.BAS" getrost vergessen.

RSX-Befehle in der Datenverwaltung

Eingabe eines RSX-Balkens erfolgt mit SHIFT+@-Taste gleichzeitig.

Syntax: |BANKOPEN,(numerischer Integerwert)

Zu Beginn eines jeden Programms muß die längste Variablengröße hiermit festgelegt werden. Sie sollten sich also Gedanken darüber machen, wie lang die längste in den zweiten 64 kByte abzulegende Stringvariable sein wird. Hierbei sind ganzzahlige Werte von 0-255 zugelassen. Null ist allerdings witzlos. Sollte bei diesem Befehl eine Fehlermeldung ausgegeben werden, ist der Bankmanager nicht oder nicht korrekt geladen. Gemäß Ihrer gewählten Länge werden die zweiten 64 kByte daraufhin organisiert. Während des Programmlaufes können Sie die Länge verändern, so oft Sie wollen. Doch Vorsicht! Daten können dabei leicht verlorengehen, wenn die Länge zu oft variiert. Es ist daher empfehlenswert, bei einer festgelegten Größe zu bleiben.

Syntax: |BANKWRITE,@(numerischer Integerwert),(Stringvariable),(wahlweise numerischer Wert)

Mit diesem Befehl werden Stringvariablen in die zweiten 64 kByte geschrieben. Hierbei muß besonders sorgfältig gearbeitet werden. Der erste Parameter muß eine Integer (ganzzahlige)-Variable sein.

Wurde zu Beginn Ihres Listings der Variablenname mit DEFINT bereits als ganzzahlig festgelegt, kann auf ein nachgestelltes %-Zeichen verzichtet werden. Ansonsten immer mit angeben! Bedeutung: Mit diesem Wert kann die als zweiter Parameter angegebene Stringvariable in die entsprechende Stelle des Speichers geschrieben werden. Allerdings muß die Variable zu

Anfang des Listings zum Beispiel mit a%=0 unbedingt definiert werden. Denn c$="Versuch":a%=3:|BANKWRITE,@a%,c$ schreibt die Stringvariable c$ in die dritte Speicherstelle. Allerdings sollten Sie es tunlichst vermeiden, diese Variable für diesen Zweck zu benutzen. Denn selbst wenn der Variablen ein Wert zugewiesen wird, heißt es noch lange nicht, daß die jeweilige Stringvariable auch an die entsprechende Stelle geschrieben wird. Dieses ist besonders dann gegeben, wenn die Variablenzuweisung innerhalb einer FOR-NEXT- oder WHILE-WEND-Schleife geschieht. Zugegeben, das kann zwei oder auch fünfzigmal gutgehen. Aber die Erfahrung hat gezeigt, daß der erste Parameter einzig und allein zur Ausgabe des Fehlercodes wirklich brauchbar ist.

UND DAS GLEICHE GILT SINNGEMÄß AUCH FÜR DEN BEFEHL |BANKREAD.

Sollte bei Einschreiben oder Auslesen eines Strings ein Fehler auftreten, wird nicht, wie sonst unter BASIC üblich, eine Fehlermeldung ausgegeben, sondern dem ersten Parameter ein negativer Wert zugewiesen.

– 1 Der gewählte Wert liegt außerhalb der zweiten RAM-Bank. Der letzte Einlese- bzw. Schreibbefehl wurde nicht ausgeführt.

– 2 Es ist ein interner Umschaltfehler bei der Auswahl des Speicherbereiches innerhalb der zweiten RAM-Bank aufgetreten.

Es ist unbedingt erforderlich, daß Sie jedesmal, wenn ein Wert ausgelesen bzw. eingeschrieben werden soll, gleich danach eine Routine aufrufen, welche prüft, ob ein Fehlercode ausgegeben wurde.

Beispiel:

100 IF a%=-1 THEN PRINT "Speicher voll! Ich gehe wieder zum Menue.": GOTO (Menü)

110 IF a%=-2 THEN CLS:PRINT"BANKUMSCHALTFEHLER! Ich versuche es noch einmal.": GOTO (zum vorherigen Einschreib- oder Auslesebefehl. Achtung! Auf dritten Parameter achten)

120 GOTO (zurück)

Eine solche Routine muß natürlich den Programmgegebenheiten angepaßt werden. Grundsätzlich ist es nach einem Bankumschaltfehler und interner Kontrolle der Variablen möglich, einen erneuten Versuch zu starten. Beim zweiten Parameter (Stringvariable) ist es erlaubt, einen normal in Anführungszeichen ausgegebenen String zu bearbeiten. Die Arbeitsweise ist die gleiche wie mit einer Stringvariablen.

Wichtig! Bevor Sie das erste Mal (und auch während des Programmablaufes beachten) arbeiten, muß eine Stringvariable erst einmal im Speicher definiert werden, sonst kann es zu Abarbeitungsfehlern kommen. Es empfiehlt sich daher, der Variablen bereits zu Beginn eines Listings die Länge des vorher mit |Bankopen,(x) festgelegten Wertes zu geben.

Nehmen wir an, Sie haben mit |Bankopen,100 die zweite RAM-Bank organisiert, und Ihre Arbeitsvariable wurde mit arbeit$ benannt. Weisen Sie nun dieser Variablen einen Leerstring oder eine andere Länge mit arbeit$=SPACE$(100) zu.

Weiterhin ist darauf zu achten, daß die Variable während des Programmablaufes diesen Wert nicht unter- beziehungsweise überschreitet. Problematik: Ist arbeit$ bei einem Schreibbefehl (|Bankwrite) kürzer und Sie haben in der gleichen Speicherstelle bereits einmal einen String eingeschrieben, werden nicht alle Zeichen überschrieben. In späteren Auslesungen kann es dann zu unliebsamen Überraschungen kommen. Ist arbeit$ bei einem Lesebefehl (|Bankread) ein Leerstring von der Länge Null, passiert (vielleicht) überhaupt nichts.

Es findet folglich keine Auslesung statt. Ist arbeit$ bei einem Schreibbefehl (|Bankwrite) länger, werden die überzähligen Zeichen einfach abgeschnitten, und es gelangt nur ein Teil des Strings in den Speicher. Dann ist es zu überlegen, ob der Wert von |Bankopen,(x) nicht vergrößert werden sollte. Vor jedem

Der Wettbewerb

Zum Sonderheft ein Sonderwettbewerb

Damit sich all die kleinen grauen Zellen unserer Hirne nach der intensiven Lektüre dieses Sonderheftes ein wenig entspannen können, wollen wir an dieser Stelle einen kleinen Wettbewerb einstreuen. Es handelt sich dabei um ein sogenanntes Spiralkreuzworträtsel. Die zu suchenden Begriffe werden von links oben im Uhrzeigersinn in das Gitter eingetragen, wobei der letzte Buchstabe eines jeden Begriffs dem ersten des nächstgesuchten entspricht. Bei richtiger Lösung ergeben die Buchstaben in den Kreisen, falls richtig zusammengesetzt, ein Lösungswort, das auf eine Postkarte geschrieben an den

DMV-Verlag
Redaktion PC Amstrad
Stichwort "Rätsel"
Postfach 250
3440 Eschwege.

geschickt werden sollte.

Der Einsendeschluß ist der 7. März 1990.

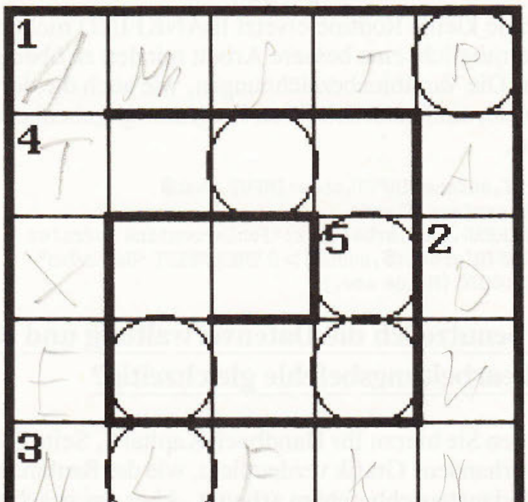
Unter allen richtigen Einsendungen werden folgende Preise verlost:

1. Preis – Ein Software-Paket für den CPC, bestehend aus Context CPC, Turbo Data CPC, Fast Basic Compiler und Copy Shop.

2. bis 10. Preis – je eines der oben genannten Programme nach eigener Wahl. Auf der Postkarte sollte daher das Wunschprogramm angegeben werden.

Der Rechtsweg ist wie immer ausgeschlossen, Mitarbeiter des DMV-Verlages und deren Angehörige dürfen leider auch diesmal nicht teilnehmen.

Aber nun zu unserem Rätsel:



1. Name der Firma, die den CPC ins Leben rief
2. Dort, wo es gut munkeln ist
3. BASIC-Befehl, der die For-Schleife beendet
4. Weiblicher Verwandter, auch als Frau des Onkels bekannt
5. Zusammensetzung aus arktischem Untergrund und brummigem braunem Artgenossen

(jf)

Zugriff auf die zweite RAM-Bank sollten Sie daher im Programm eine Routine einrichten, die prüft, ob der String zu lang oder zu kurz ist (sofern dieses nicht vorher geschehen ist).

Beispiel:
10 länge=100:|Bankopen,100 usw......
100 IF LEN(arbeit$) > länge THEN PRINT "STRING TOO LONG-ERROR!":GOTO (zurück zur Eingabe)
110 WHILE LEN(arbeit$) < länge: arbeit$ = arbeit$+" ": WEND:GOTO (Lese/Schreibbefehl)

Der dritte Parameter sollte nach Möglichkeit ein Integerwert sein. Gemäß meiner vorher gemachten Ausführungen ist der Parameter eigentlich kein Wahlparameter mehr, sondern ein MUSS.

Der Wert darf zwischen 0 und .... hängt von der Speicherorganisation durch |Bankopen,(x) ab (siehe auch Handbuch, Kap.8, S.4 bis 6).

Syntax: |Bankread,@(numerischer Integerwert), (Stringvariable),(wahlweise numerischer Wert)
Für |Bankread gilt sinngemäß das gleiche, wie zu |Bankwrite ausgeführt. Nur daß hier ausschließlich Stringvariablen aus der RAM-Bank gelesen werden.

Syntax: |Bankfind,@(numerischer Integerwert),(Text), (numerischer Integerwert BEGINN), (numerischer Integerwert ENDE)
Dieser Befehl ist ähnlich dem INSTR-Befehl des normalen BASIC. Auch hier sollte der erste Parameter nur zur Überprüfung genutzt werden. Ansonsten vergleichen Sie die Hinweise in Ihrem Handbuch in Kapitel 8, Seite 6+7.

Leider, der Befehl hat so seine Macken!

1. Leicht kommt es bei Nutzung zu einem Fehlercode (häufiger -2).
2. Auch der zu suchende String muß auf die Höchstlänge eines Datensatzes gebracht werden.
3. Die Benutzung ist nicht komfortabel.
4. Sie sind von Groß- und Kleinschreibung abhängig.

Ihr CPC 6128 stellt für diesen Zweck einen wesentlich mächtigeren Befehl zur Verfügung (INSTR).

Folgende kleine Routine ersetzt |BANKFIND nicht nur, sondern ermöglicht eine bessere Arbeit mit den zu übergebenden Daten. Die Variablenbezeichnungen, wie auch die der anderen Beispiele, sind natürlich Ihren Programmgegebenheiten anzupassen.

```
10 INPUT,anfang:INPUT,ende:INPUT,such$
20 FOR x=anfang TO ende
30 |BANKREAD,@a%,arbeit$,x:'Fehlerroutine aufrufen
40 IF INSTR(arbeit$,such$)>0 THEN PRINT "Gefunden"
50 NEXT:GOTO (Menue usw.)
```

Wie benutze ich die Datenverwaltung und die Bildbearbeitungsbefehle gleichzeitig?

Schlagen Sie hierzu Ihr Handbuch Kapitel 8, Seite 2, auf. Die dort vorhandene Grafik verdeutlicht, wie der Bankmanager mit den Bildaustauschbefehlen arbeitet. Aber das geschieht nicht nur bei |Screenswap oder |Screencopy, sondern auch bei der Datenverwaltung mit den anderen Befehlen (nur unmerklich). Mit den Datenverwaltungsbefehlen werden die zweiten 64 kByte immer von Block 4 bis 7 in aufsteigender Reihenfolge gefüllt.

Der Vorteil der Bildaustauschbefehle liegt auf der Hand. Hier können Sie selbst anhand der Parameter bestimmen, welcher Block gefüllt wird. Wenn Sie nun beide Möglichkeiten nutzen wollen, müssen Sie durch einfache Berechnungen nur sicherstellen, daß sich die Befehle "nicht in die Quere" kommen.

Beispielprogramm Zeileneditor

Genug der Theorie. Für alle, die auch gerne ein praktisches Beispiel haben, ist das Listing BANKMAN2.BAS von Interesse. Gegenüber den anderen Rechnern der CPC-Reihe bietet es erhebliche Arbeitserleichterungen und Geschwindigkeitsgewinne. Es handelt sich hierbei um einen einfachen Zeileneditor für BASIC-Programme. Mit etwas Programmiererfahrung können Sie ihn problemlos erweitern (Beispiel: Erkennungsroutine für BASIC-Schlüsselwörter). Wer möchte, kann hier mit MERGE das BANKMAN1-Listing bereits einfügen.

Bedienung:

Das zu bearbeitende Programm muß mit SAVE"a",a als ASCII-Datei abgespeichert werden. Nach Möglichkeit sollte das zu bearbeitende Programm höchstens aus 256 Zeilen bestehen und vor Abspeicherung mit RENUM eine Zeilenorganisation durchgeführt werden (muß nicht). Das Programm besteht hauptsächlich aus fünf Optionen, welche während des Programmlaufes fast immer bereit sind. Gerade begonnene Optionen werden dann allerdings nicht mehr durchgeführt. Alle Optionen werden mit den Tasten des Zehnerblockes aufgerufen. Aus Platzgründen wird hier nur auf die wesentlichen Optionen eingegangen.

<f1> Zeileneditor: Als Vorzeichen ist vor jeder Nummer entweder ein "d" für einen bekannten und bestimmten Datensatz, oder, falls nicht bekannt, ein "z" für Zeilennummer einzugeben. Bei "d" ist die Abarbeitung allerdings schneller, daher auch das RENUM vor Abspeicherung. Faustregel bei eingeladenen Programmen mit Zeilennummernbeginn bei 10 und 10 Zeilen Unterschied zur nächsten: Zeilennummer / 10 = Datensatz. Wird die entsprechende Zeile gefunden, werden, je nach Länge der vorhergehenden Zeilen, bis zu fünf gleich zum Nachschauen mit auf dem Bildschirm ausgegeben. Geändert wird genauso, wie bei dem Ihnen bereits bekannten String-Editor. Als kleines Schmankerl können Sie aus den abgebildeten Zeilen auch mit der COPY/CURSOR-Methode einzelne Zeichen kopieren.

<f2> Zeichenfolge suchen: Hier kann eine zu suchende Zeichenfolge (auch Steuercodes mit der CTRL-Tastenmethode) mit einer maximalen Länge von 255 Zeichen eingegeben werden. Übrigens! Das Programm verarbeitet in jedem Falle über CONTROL eingegebene Steuerzeichen. Auf Groß- und Kleinschreibung braucht nicht geachtet zu werden. Es sei denn, Sie geben ein BASIC-Schlüsselwort ein.

<f3> Zeilen zeigen: Ist dem LIST-Befehl im Direktmodus ähnlich. Eingaben müssen hier allerdings immer Datensatznummern sein, keine Zeilen.

<f4> Beenden: Der RAM-Bank-Inhalt wird unter dem Namen "A" abgespeichert und gleich gestartet. Die alte "A"-Datei erhält den Namen "A.BAK". Besonderheit: Der Speicherinhalt der RAM-Bank bleibt bestehen. Nach Neustart des Programms können Sie dann diesen Inhalt weiterbearbeiten. Aus Gründen erhöhter Arbeitsgeschwindigkeit wird unter dem Namen "B" immer eine zusätzliche Datei abgespeichert, um die enthaltenen Daten bei Programmbeginn nicht neu zu berechnen. Diese muß immer mit auf der Diskette mit der dazugehörenden "A"-Datei vorhanden sein.

<f5> Neustart: Das Programm wird neu gestartet. Der Inhalt der RAM-Bank ist dennoch weiter bearbeitbar.

(Holger Schäkel/cd)

für 6128

```
10 ' *********************************** [1383]
20 ' *        Listing Bankmanl.Bas      * [1940]
30 ' *********************************** [1383]
40 '                                      [117]
50 IF HIMEM=42619 AND PEEK(41431)<>201 THE [15355]
N SYMBOL AFTER 256:MEMORY 41430:LOAD"bankm
an.bin",41431:SYMBOL AFTER 240:CALL 41431:
ELSE IF PEEK(41431)<>201 THEN PRINT"Speich
erorganisation nicht kompatibel. Neustart
erbeten!":END
```

```
10 ' Als Zeile 10 koennen Sie hier das Lis [3866]
ting BANKMAN1.BAS nutzen.
20 ' *********************************** [1543]
*
30   a$="Programmbearbeitung auf CPC 6128  [3493]
"
40   b$=" von Holger Schaekel"             [1792]
50 ' *********************************** [1543]
*
60 ' *** Initialisierungen                [1283]
70 '                                       [117]
80 ON ERROR GOTO 2140                     [1471]
90 MODE 2:DEFINT a-z:DIM zahl(257),nummer( [8345]
257):a=0:an=1:anfang=0:ende=0:INK 0,26:INK
 1,0:PEN 1:PEN#1,0:PAPER 0:PAPER#1,1:CLS:B
ORDER 26
100 ¦BANKOPEN,255:lese$=SPACE$(255):c$="<f [6284]
1> Zeilenedit <f2> Zeichensuche <f3> Zeile
n zeigen <f4> Beenden <f5> Neustart "
110 PRINT STRING$(81,42)SPC(8)CHR$(24)a$CH [4678]
R$(164)" 05/1988"b$CHR$(24)SPC(8)"*"STRING
$(80,42)
120 FOR x=4 TO 23:LOCATE 1,x:PRINT"*":LOCA [4932]
TE 80,x:PRINT"*":NEXT
130 LOCATE 1,24:PRINT CHR$(24)SPC(28)"* Ak [5329]
tuelle Menuepunkte *"SPC(28)CHR$(24)
140 WINDOW 3,78,5,23:WINDOW#1,1,80,25,25:W [10464]
INDOW#2,3,78,5,23:CLS:CLS#1:PRINT"Bitte Di
skette mit Datei >"CHR$(24)"A"CHR$(24)"< e
inlegen!"
150 PRINT:PRINT"Ist das gewuenschte Progra [5879]
mm bereits in der zweiten RAM-Bank [J/N]"
160 GOSUB 1140:CLS#1:IF a$="J" THEN GOSUB  [2871]
1230:IF daten>1 THEN 300
170 ' *** Programm laden ***              [2253]
180 '                                      [117]
190 PRINT:PRINT"Bitte eine Taste druecken" [4852]
:CLEAR INPUT:CALL &BB06
200 PRINT CHR$(11)SPC(25):daten=1:fehler=0 [4352]
:¦ERA,"b":PRINT CHR$(11)CHR$(20)
210 OPENIN "A":OPENOUT"B":WHILE NOT EOF    [2439]
220 LINE INPUT#9,lese$:IF NOT EOF AND(lese [5124]
$="" OR lese$=SPACE$(255))THEN 220
230 zahl(daten)=LEN(lese$):nummer(daten)=V [6150]
AL(lese$):WRITE#9,zahl(daten),nummer(daten
):GOSUB 1190
240 ¦BANKWRITE,@a,lese$,daten:IF a=-2 THEN [3444]
 ERROR 33
250 IF a=-1 THEN PRINT:PRINT"Speicher voll [4688]
!":GOSUB 1370:GOTO 280
260 LOCATE 1,5:PRINT"Ich bearbeite Datensa [4490]
tz :";USING"####";daten
270 daten=daten+1:WEND                    [1280]
280 daten=daten-1:CLOSEIN:CLOSEOUT        [2388]
290 ¦BANKWRITE,@a,DEC$(daten,"####")+SPACE [4806]
$(246),0:IF a=-2 THEN ERROR 33
300 ' *** Laenge laden ***                [2652]
310 '                                      [117]
```

Listing Bankmanager

```
320 IF a$="J" THEN OPENIN"B":FOR x=1 TO da [4442]
ten:INPUT#9,zahl(x),nummer(x):NEXT:CLOSEIN
330 ' *** Hauptroutine ***                [1840]
340 '                                      [117]
350 LOCATE#3,2,24:PRINT#3,CHR$(24)"Daten : [3338]
"DEC$(daten,"####")CHR$(24)
360 CLS:CLS#1:CLEAR INPUT:pruef$="":x=80:s [2300]
pringe=10:kopie=0
370 WHILE springe=10:FRAME                [1607]
380 LOCATE#1,1,1:PRINT#1,RIGHT$(c$,x)LEFT$ [2379]
(c$,80-x);
390 GOSUB 1290:x=x-1:IF x=0 THEN x=80     [1530]
400 WEND:PRINT#1,c$;                       [953]
410 ON springe GOTO 420,650,730,890,1020  [1921]
420 ' *** <f1> Zeilenedit ***              [988]
430 '                                      [117]
440 kopie=1:fehler=0:datenl=0:CURSOR 0,0:C [3170]
LS
450 PRINT"Welche(r) "CHR$(24)"D"CHR$(24)"a [6044]
tensatz oder "CHR$(24)"Z"CHR$(24)"eile :"
460 PRINT:PRINT"Eingabe mit Vorzeichen : " [5759]
;:laenge=6:GOSUB 1980:such$=hilf$
470 IF springe<>10 THEN 410               [2148]
480 IF such$="" THEN 330                  [1472]
490 pruef$=UPPER$(LEFT$(such$,1))         [2039]
500 IF pruef$="Z" THEN such$=RIGHT$(such$, [3668]
LEN(such$)-1):anfang=1:ende=daten:GOTO 153
0
510 IF pruef$="D" THEN datenl=VAL(RIGHT$(s [1826]
uch$,LEN(such$)-1))
520 IF datenl<1 OR datenl>daten THEN PRINT [6050]
:PRINT"Falsche Werte. Bitte neu auswaehlen
!":GOSUB 1370:GOTO 420
530 CLS:anfang=1:zahll=0:IF datenl-anfang< [4259]
=1 THEN anfang=1:GOTO 600
540 zahll=zahll+zahl(datenl-anfang):IF zah [5549]
ll<537 AND datenl-anfang>1 THEN anfang=anf
ang+1:GOTO 540
550 IF datenl-anfang<1 THEN anfang=anfang- [2658]
1:GOTO 550
560 IF anfang>5 THEN anfang=5             [1605]
570 FOR x=anfang TO 1 STEP-1              [1625]
580 ¦BANKREAD,@a,lese$,datenl-x:IF a=-2 TH [3038]
EN ERROR 33
590 GOSUB 1870:NEXT                       [1502]
600 ¦BANKREAD,@a,lese$,datenl:IF a=-2 THEN [2723]
 ERROR 33
610 zaehler=datenl:GOSUB 1650:IF springe<> [3402]
10 THEN 410
620 GOSUB 2090                             [865]
630 ¦BANKWRITE,@a,lese$,datenl:IF a=-2 THE [3229]
N ERROR 33
640 GOTO 330                               [506]
650 ' *** <f2> Zeichen suchen ***         [1626]
660 '                                      [117]
670 pruef$="":fehler=0:datenl=1:CURSOR 0,0 [2820]
:CLS
680 PRINT"Bitte geben Sie die zu suchende  [5635]
Zeichenfolge ein :"
690 PRINT:PRINT:laenge=255:GOSUB 1980:such [4309]
$=hilf$
700 IF springe<>10 THEN 410               [2148]
710 IF such$="" THEN 330                  [1472]
720 GOSUB 1420:GOTO 330                   [1368]
730 ' *** <f3> Zeilen zeigen ***          [1207]
740 '                                      [117]
750 anfang=1:ende=daten:fehler=0:CURSOR 0, [2261]
0:CLS
760 anfang=1:LOCATE 1,3:PRINT"Beginn Daten [6224]
 : ";:laenge=6:GOSUB 1980:IF hilf$="" THEN
 780 ELSE anfang=VAL(hilf$)
770 IF anfang<1 OR anfang>daten THEN 760  [2841]
780 IF springe<>10 THEN 410               [2148]
790 ende=daten:LOCATE 1,5:PRINT"Ende Daten [5034]
   : ";:laenge=6:GOSUB 1980:IF hilf$="" TH
EN 810 ELSE ende=VAL(hilf$)
800 IF ende<anfang OR ende>daten THEN 790 [2535]
810 IF springe<>10 THEN 410               [2148]
820 FOR x=anfang TO ende                  [1553]
830 ¦BANKREAD,@a,lese$,x:IF a=-2 THEN ERRO [2837]
R 33
840 LOCATE 1,7:PRINT"Arbeitsdaten : ";USIN [3180]
G"####";x
850 zaehler=x:GOSUB 1650:IF springe<>10 TH [2658]
EN 410
860 GOSUB 2090                             [865]
870 ¦BANKWRITE,@a,lese$,x:IF a=-2 THEN ERR [4290]
OR 33
880 NEXT:GOTO 330                         [1172]
```

Listing Bankmanager

```
890 ' *** <f4> Beenden ***                   [514]
900 '                                        [117]
910 fehler=0:CURSOR 0,0:CLS:PRINT"Soll das [5869]
 Programm wirklich beendet werden [J/N]":G
OSUB 1140
920 IF a$="N" THEN 330                   [937]
930 |ERA,"a.bak":|REN,"a.bak","a":OPENOUT" [4316]
a"
940 FOR x=1 TO daten:lese$=SPACE$(255)   [2728]
950 LOCATE 1,3:PRINT"Datensatz  : ";USING" [2650]
####";x
960 |BANKREAD,@a,lese$,x:IF a=-2 THEN ERRO [2848]
R 34
970 IF lese$=SPACE$(255) THEN 1000       [1963]
980 lese$=LEFT$(lese$,zahl(x))           [1629]
990 PRINT#9,lese$                        [1265]
1000 NEXT:CLOSEOUT:GOSUB 1090            [1596]
1010 MODE 2:CALL &BC02:RUN"a"            [1583]
1020 ' *** <f5> Neustart ***             [1053]
1030 '                                   [117]
1040 fehler=0:CURSOR 0,0:CLS:PRINT"Sie wue [5628]
nschen einen Neustart [J/N]":GOSUB 1140
1050 IF a$="N" THEN 330                  [937]
1060 GOSUB 1090:RUN                      [1556]
1070 ' *** Unterprogramme ***            [1727]
1080 '                                   [117]
1090 ' *** Laenge sichern ***            [2861]
1100 '                                   [117]
1110 |ERA,"b":OPENOUT"b":FOR x=1 TO daten [3569]
1120 WRITE#9,zahl(x),nummer(x)           [1731]
1130 NEXT:CLOSEOUT:RETURN                [855]
1140 ' *** [J/N] Abfrage ***             [1131]
1150 '                                   [117]
1160 CLS#1:CLEAR INPUT:a$=""             [907]
1170 WHILE a$<>"J" AND a$<>"N":a$=UPPER$(I [4071]
NKEY$):WEND
1180 RETURN                              [555]
1190 ' *** Stringlaenge ***              [1019]
1200 '                                   [117]
1210 laenge=LEN(lese$):IF laenge<255 THEN [3583]
lese$=lese$+STRING$(255-laenge,32)
1220 RETURN                              [555]
1230 ' *** Programm bereits im Speicher ** [2529]
*
1240 '                                   [117]
1250 |BANKREAD,@a,lese$,0:IF a=-2 THEN ERR [1996]
OR 33
1260 daten=VAL(lese$)                    [253]
1270 IF daten<1 THEN PRINT:PRINT"Stimmt ja [4237]
 gar nicht!":a$="N":GOSUB 1370
1280 RETURN                              [555]
1290 ' *** Welche Taste wurde gedrueckt ** [3180]
*
1300 '                                   [117]
1310 springe=10:IF INKEY(13)=0 THEN spring [3946]
e=1
1320 IF INKEY(14)=0 THEN springe=2       [905]
1330 IF INKEY(5)=0 THEN springe=3        [981]
1340 IF INKEY(20)=0 THEN springe=4       [1652]
1350 IF INKEY(12)=0 THEN springe=5       [1563]
1360 RETURN                              [555]
1370 ' *** Warteschlange ***             [1745]
1380 '                                   [117]
1390 pause=1:AFTER 150,0 GOSUB 1410:PRINT [2929]
CHR$(7)
1400 WHILE pause=1:WEND                  [1958]
1410 pause=0:RETURN                      [2026]
1420 ' *** Zeichenfolge suchen ***       [3158]
1430 '                                   [117]
1440 anfang=1:ende=daten:fehler=0        [1236]
1450 CLS:PRINT"Soll bestimmbar bei der Suc [7211]
he vorgegangen werden [J/N]":GOSUB 1140:IF
 a$="N" THEN 1520
1460 anfang=1:LOCATE 1,5:PRINT"Anfang Such [7173]
e bei :";:laenge=6:GOSUB 1980:IF hilf$=""
THEN 1480 ELSE anfang=VAL(hilf$)
1470 IF anfang<1 OR anfang>daten THEN 1460 [2990]
1480 IF springe<>10 THEN 410             [2148]
1490 ende=daten:LOCATE 1,7:PRINT"Ende der [7234]
Suche bei :";:laenge=6:GOSUB 1980:IF hilf$
="" THEN 1510 ELSE ende=VAL(hilf$)
1500 IF ende<1 OR ende>daten OR ende<anfan [2133]
g THEN 1490
1510 IF springe<>10 THEN 410             [2148]
1520 CLS:PRINT CHR$(24)"Gesucht wird :"CHR [7712]
$(24):PRINT:FOR y=1 TO LEN(such$):PRINT CH
R$(1)MID$(such$,y,1);:NEXT
```

Listing Bankmanager

```
1530 PRINT#1,c$;:FOR x=anfang TO ende     [2547]
1540 IF pruef$="" THEN LOCATE 1,8:PRINT"Da [4808]
tensatz : ";USING"####";x
1550 |BANKREAD,@a,lese$,x:IF a=-2 THEN ERR [2837]
OR 33
1560 GOSUB 1290:IF springe<>10 THEN 410   [1877]
1570 IF pruef$="Z" AND VAL(lese$)=VAL(such [1022]
$) THEN 1600
1580 IF pruef$="" AND INSTR(UPPER$(lese$), [2829]
UPPER$(such$))>0 THEN 1610
1590 NEXT:CLS:PRINT"Ich habe das Ende der  [7878]
Daten erreicht!":GOSUB 1370:GOTO 330
1600 IF pruef$="Z" THEN daten1=x:GOTO 530  [1041]
1610 zaehler=x:GOSUB 1650:IF springe<>10 T [2658]
HEN 410
1620 GOSUB 2090                           [865]
1630 |BANKWRITE,@a,lese$,x:IF a=-2 THEN ER [4290]
ROR 33
1640 GOTO 1590                            [317]
1650 ' *** Editor ***                     [1227]
1660 '                                    [117]
1670 falsch=1:LOCATE 1,15:GOSUB 1870:LOCAT [8293]
E 1,19:PRINT SPC(43);:LOCATE 1,15:buchstab
e=1:PRINT CHR$(7);:CURSOR 1,1
1680 CLEAR INPUT:tipp$="":WHILE tipp$="":t [3222]
ipp$=INKEY$:WEND
1690 li=POS(#0):re=VPOS(#0):lin=POS(#2):re [3895]
c=VPOS(#2):ti=ASC(tipp$)
1700 GOSUB 1290:IF ti=13 OR springe<>10 TH [18819]
EN CURSOR 0,0:CURSOR#2,0,0:an=1:GOSUB 1930
:IF(VAL(lese$)<>0 OR lese$=SPACE$(255))AND
 falsch THEN nummer(zaehler)=VAL(lese$):RE
TURN:ELSE LOCATE 1,19:PRINT"Zeilennummer f
ehlt bzw. bereits vorhanden!":GOTO 1650
1710 IF kopie AND an AND(ti=247 OR ti=246  [7667]
OR ti=244 OR ti=245)THEN an=0:ti=255:LOCAT
E#2,li,re:CURSOR#2,1,1
1720 IF ti=243 THEN ti=255:IF buchstabe<25 [6064]
5 THEN PRINT CHR$(9);:buchstabe=buchstabe+
1:GOTO 1680
1730 IF ti=247 AND kopie THEN ti=255:IF li [5563]
n+rec<>95 THEN PRINT#2,CHR$(9);
1740 IF ti=242 THEN ti=255:IF buchstabe>1  [4816]
THEN PRINT CHR$(8);:buchstabe=buchstabe-1:
GOTO 1680
1750 IF ti=246 AND kopie THEN ti=255:IF li [3484]
n+rec<>2 THEN PRINT#2,CHR$(8);
1760 IF ti=240 THEN ti=255:IF re>15 THEN P [5842]
RINT CHR$(11);:buchstabe=buchstabe-76:GOTO
 1680
1770 IF ti=244 AND kopie THEN ti=255:IF re [2609]
c>1 THEN PRINT#2,CHR$(11);
1780 IF ti=241 THEN ti=255:IF re<18 THEN I [7673]
F buchstabe+76>255 THEN 1680 ELSE PRINT CH
R$(10);:buchstabe=buchstabe+76:GOTO 1680
1790 IF ti=245 AND kopie THEN ti=255:IF re [3895]
c<19 THEN PRINT#2,CHR$(10);
1800 IF ti=127 THEN lese$=LEFT$(lese$,buch [8678]
stabe-1)+RIGHT$(lese$,255-buchstabe)+" ":C
URSOR 0,0:LOCATE 1,15:GOSUB 1870:LOCATE li
,re:CURSOR 1,1:GOTO 1680
1810 IF INKEY(47)=32 AND buchstabe>1 THEN  [14095]
lese$=LEFT$(lese$,buchstabe-1)+" "+MID$(le
se$,buchstabe,256-(buchstabe+1)):CURSOR 0,
0:LOCATE 1,15:GOSUB 1870:LOCATE li,re:CURS
OR 1,1:GOTO 1680
1820 IF ti=16 THEN lese$=SPACE$(255):CURSO [4516]
R 0,0:LOCATE 1,15:PRINT lese$;:LOCATE 1,15
:CURSOR 1,1:buchstabe=1:GOTO 1680
1830 IF ti=224 THEN tipp$=" "             [1104]
1840 IF ti=224 AND kopie AND an=0 THEN tip [5519]
p$=COPYCHR$(#2):IF lin+rec<>95 THEN PRINT#
2,CHR$(9);
1850 IF ti=255 THEN 1680 ELSE IF buchstabe [4169]
>255 THEN buchstabe=255:PRINT CHR$(8);:GOT
O 1680
1860 PRINT CHR$(1)tipp$;:MID$(lese$,buchst [6579]
abe,1)=tipp$:buchstabe=buchstabe+1:GOTO 16
80
1870 ' *** Zeile ausgeben ***             [2939]
1880 '                                    [117]
1890 FOR y=1 TO 255:PRINT CHR$(1)MID$(lese [3041]
$,y,1);
1900 IF RIGHT$(lese$,255-y)=SPACE$(255-y)T [3608]
HEN y=255:PRINT CHR$(20)
1910 NEXT                                 [350]
1920 PRINT:RETURN                         [914]
```

Listing Bankmanager

```
1930 '*** Nummer pruefen ***                   [1108]
1940 '                                         [117]
1950 FOR y=1 TO daten                          [1413]
1960 IF nummer(y)=VAL(lese$)AND zaehler<>y [2624]
 THEN y=daten:falsch=0
1970 NEXT:RETURN                               [940]
1980 ' *** Eingabe ***                         [1674]
1990 '                                         [117]
2000 CLEAR INPUT:hilf$="":CURSOR 1,1           [1509]
2010 a$="":WHILE a$="":a$=INKEY$:WEND          [1812]
2020 GOSUB 1290                                [865]
2030 IF a$=CHR$(13) OR springe<>10 THEN CU [1217]
RSOR 0,0:PRINT:RETURN
2040 IF INKEY(16)=0 THEN CURSOR 0,0:x=LEN( [4754]
hilf$):PRINT STRING$(x,8)SPC(x)STRING$(x,8
);:GOTO 2000
2050 IF INKEY(79)=0 AND LEN(hilf$)>0 THEN  [5523]
PRINT CHR$(8)" "CHR$(8);:hilf$=LEFT$(hilf$
,LEN(hilf$)-1):GOTO 2010
2060 IF NOT(INKEY(2) AND INKEY(0) AND INKE [4627]
Y(1) AND INKEY(8) AND INKEY(79)) THEN 2010
2070 IF LEN(hilf$)=laenge THEN 2010            [2316]
2080 PRINT a$;:hilf$=hilf$+a$:GOTO 2010        [2460]
2090 ' *** Variable zahl? ***                  [1683]
2100 '                                         [117]
2110 FOR z=1 TO 255                            [1142]
2120 IF RIGHT$(lese$,255-z)=SPACE$(255-z)T [4529]
HEN zahl(zaehler)=255-(255-z):z=255
2130 NEXT:RETURN                               [940]
2140 ' *** Fehler abfangen ***                 [1441]
2150 '                                         [117]
2160 CLS:diskfehler=DERR AND &7F:fehler=fe [1595]
hler+1
2170 IF ERR<33 OR fehler>5 THEN 2290           [1772]
2180 IF ERL=240 THEN RESUME 240                [1404]
2190 IF ERL=290 THEN RESUME 290                [2088]
2200 IF ERL=580 THEN RESUME 580                [1138]
```

Listing Bankmanager

```
2210 IF ERL=600 THEN RESUME 600                [319]
2220 IF ERL=630 THEN RESUME 630                [1016]
2230 IF ERL=830 THEN RESUME 830                [974]
2240 IF ERL=870 THEN RESUME 870                [1157]
2250 IF ERL=960 THEN RESUME 960                [1745]
2260 IF ERL=1250 THEN RESUME 1250              [1473]
2270 IF ERL=1550 THEN RESUME 1550              [1038]
2280 IF ERL=1630 THEN RESUME 1630              [1497]
2290 IF ERR>32 THEN PRINT"'eider ist ein u [5499]
nreparabler BANKUMSCHALTFEHLER aufgetaucht
!"
2300 IF ERR=33 THEN PRINT:PRINT"Ich versuc [10722]
he durch Abspeicherung, die Daten zu rette
n!":PRINT:PRINT CHR$(24)"TASTE"CHR$(24):CL
EAR INPUT:CALL &BB06:RESUME 890
2310 IF ERR=34 THEN PRINT:PRINT"Ich versuc [10694]
he durch Neustart, die Daten zu retten!":P
RINT:PRINT CHR$(24)"TASTE"CHR$(24):CLEAR I
NPUT:CALL &BB06:GOSUB 2310:RUN
2320 IF diskfehler=19 OR diskfehler=20 THE [9631]
N PRINT"Diskette ist voll. Bitte eine ande
re einlegen. Programm wird neugestartet!":
GOSUB 1370:RUN
2330 PRINT"Es ist ein Fehler aufgetaucht!" [2811]
2340 PRINT:PRINT"In Zeile        : "DEC$(E [2258]
RL,"####")
2350 PRINT:PRINT"Fehlernummer    : "DEC$(E [3024]
RR,"####")
2360 PRINT:PRINT"Diskettenfehler : "DEC$(d [2437]
iskfehler,"####")
2370 PRINT:PRINT"Bitte schauen Sie in Ihre [3768]
m Handbuch nach."
2380 PRINT:PRINT CHR$(24)"Programm wurde b [2855]
eendet!"CHR$(24)
2390 IF ERR=28 THEN PRINT:PRINT"BANKMAN no [2622]
ch laden!"
2400 PRINT:END                                 [738]
```

Listing Bankmanager

Sie wollen uns ein Programm zusenden! Folgende Regeln sollten Sie beachten:

1) Schicken Sie Ihre Programme auf Diskette oder Kassette. Dabei mehrmals abspeichern.

2) Legen Sie eine ausführliche Programmbeschreibung bei, aus der eindeutig der Sinn und Zweck des Programms hervorgeht. Am besten speichern Sie die Beschreibung als Textfile (Wordstar, ConText, LocoScript o.ä.) ebenfalls auf Datenträger.

3) Schicken Sie auf jeden Fall ein Programmlisting mit.

4) Vergessen Sie Ihren Absender nicht! Vollständige Anschrift, wenn möglich auch die Telefonnummer, ersparen unnötige Verzögerungen unsererseits.

5) Behalten Sie auf jeden Fall eine Kopie Ihres Programms, unvorhersehbare Umstände werden somit nicht zur Katastrophe.

6) Wenn Sie sogar mehrere Programme geschrieben haben und diese veröffentlichen wollen, nehmen Sie bitte für jedes Programm einen separaten Datenträger (mehrmals abspeichern!).

Damit erleichtern Sie uns die Arbeit wesentlich, und eine schnellere Bearbeitung wird möglich.

7) Versehen Sie das Programm mit den sehr wichtigen REM-Statements. Das fördert nicht nur die Übersicht für uns und unsere Leser, sondern unterstützt zusätzlich das strukturierte Programmieren. Merke: Auch erfahrene Programmierer verwenden REM-Zeilen, das Programm sollte schließlich jeder verstehen und anwenden können.

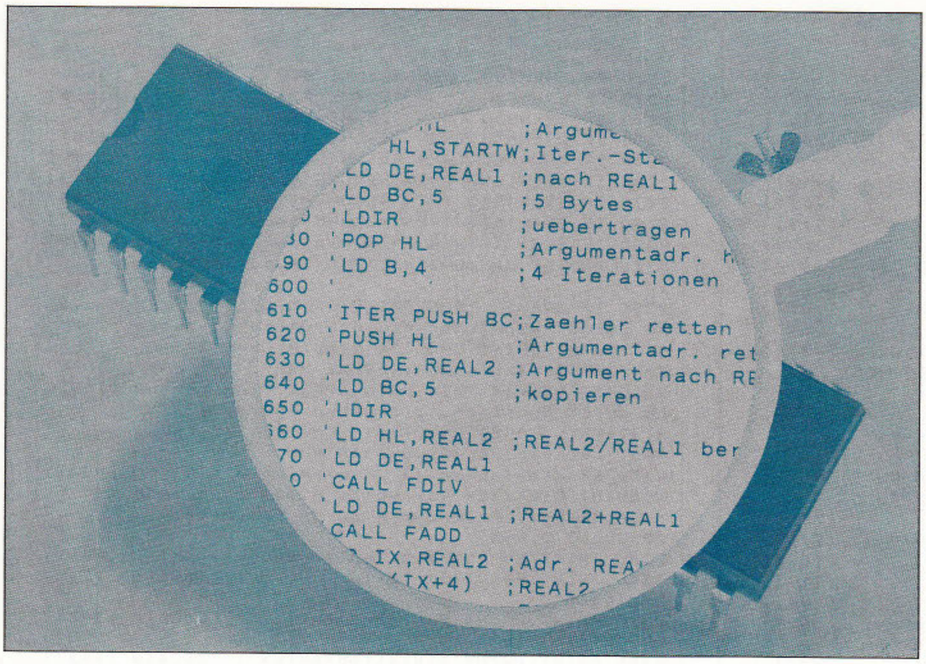
8) Vermeiden Sie möglichst Variablennamen, wo leicht I und i oder O und o verwechselt werden können. Variablen sollten grundsätzlich in Kleinbuchstaben geschrieben werden und möglichst verständlich sein. Die Variable butter sagt mehr aus als nur bu.

9) Bieten Sie das Programm niemals verschiedenen Verlagen gleichzeitig an! Mit der Veröffentlichung und dem daraus resultierenden Honorar gehen die Urheberrechte an den Verlag über! Sollte Ihr Programm noch in einer anderen Zeitschrift abgedruckt werden, liegt ein Verstoß gegen das Wettbewerbsrecht vor.

10) Jedes bei uns ordnungsgemäß eingegangene Programm wird ausführlich begutachtet und getestet. Sie erhalten von uns dann umgehend Bescheid, ob Ihr Programm veröffentlicht wird oder nicht. Wenn Sie das Programm nach unserer Begutachtung zurückhaben wollen, legen Sie bitte der Einsendung einen frankierten Rückumschlag bei.

Falls wir uns für den Abdruck Ihres Programms entschieden haben, setzen wir uns unverzüglich mit Ihnen in Verbindung und besprechen Ihr verdientes Honorar!
Übrigens wäre es nett, wenn Sie uns auch ein paar persönliche Daten schreiben würden (z.B. Ihr Alter, Tätigkeit, wie und wann kam die Idee zu dem Programm, was alles machen Sie mit dem Rechner usw.?). So, jetzt aber genug der Theorie, mit fast unerträglicher Spannung und Neugier wartet Ihre »PC International« auf Ihre Beiträge.

Einsendung bitte an: DMV-Verlag · Postfach 250 · 3440 Eschwege



Die Prozessor-Lupe

Einzelschrittsimulator für CPC 464

Viele Leser, die in Maschinensprache programmieren, werden folgende Situation sicherlich kennen: Das mühsam erstellte Programm stürzt schon beim ersten Probelauf ab, und eine langwierige Fehlersuche beginnt. Dieses Debugging kann mit einem Einzelschrittsimulator, der die Abarbeitung des Maschinenprogrammes simuliert und nach jedem Befehl die Registerinhalte anzeigt, erheblich verkürzt werden.

Die Programmübersicht

- Die Verarbeitung fast aller Z80-Befehle.
- Eine übersichtliche Ausgabe aller Registerinhalte nach jedem Schritt.
- Die Möglichkeit, sowohl im echten Einzelschrittmodus zu arbeiten (das heißt, das Programm wartet nach jedem Schritt auf einen Tastendruck), oder aber eine fortlaufende Abarbeitung der Befehle zu wählen.
- Eine hohe Verarbeitungsgeschwindigkeit durch Programmierung in Maschinensprache.
- Eine einfache Bedienung.

Der Umgang mit dem Simulator

Doch nun zum Programm selbst. Nachdem das Programm geladen ist, werden Sie nach der Startadresse des zu testenden Programmes gefragt. Es muß unterhalb der Adresse &9C50 liegen, da der Binärteil des Simulators im Adreßbereich ab &9C50 liegt. Ferner wäre es auch nützlich, wenn Sie neben sich ein Listing Ihres Maschinenprogrammes liegen hätten. Danach drücken Sie bei der Abfrage "BASIC-Aufruf (j/n) ?" die n-Taste. Der Bildschirm wird nun gelöscht, das Programm simuliert den ersten Befehl, den es an Ihrer Startadresse findet und zeigt die Registerinhalte in der unteren Bildschirmhälfte an. Drücken Sie nun irgendeine Taste außer ENTER und das Programm nimmt sich den nächsten Befehl vor. Am Inhalt des PC-Registers erkennen Sie, wo der Befehl steht, der als nächstes simuliert wird. Diesen Vorgang können Sie nun solange wiederholen, bis die Meldung *RETURN* erscheint. Nach einem erneuten Tastendruck kehrt das Programm in Zeile 440 des BASIC-Programmes zurück. Wenn Sie nun die ENTER-Taste drücken, so wird das Programm beendet. Andernfalls werden Sie wieder nach einer Startadresse gefragt und die Simulation beginnt von neuem. Mit ENTER können Sie die Simulation jederzeit unterbrechen und in Zeile 440 zurückkehren.

Der BASIC-Aufruf

Wenn Sie bei der "BASIC-Aufruf (j/n) ?" Abfrage j eingeben, also BASIC-Aufruf wählen, so können noch bis zu 32 Parameter übergeben werden. Sie werden dann vom Programm nach der Anzahl der Parameter gefragt. Wenn Sie einen BASIC-Aufruf ohne Parameter wünschen, geben Sie 0 ein, ansonsten die Anzahl der zu übergebenden Parameter. Anschließend können Sie die Parameter eingeben. Bei Beginn der Simulation haben dann die einzelnen Register folgenden Inhalt:

ohne Parameter	**mit Parametern**
A : 0	*n*
F : &68	*&28*
B : &20	*&20 - n*
C : &FF	*&FF*
DE: anzuspringende Adresse	*letzter übergebener Parameter*
IX: Stapeladr. &BFFE	*Stapeladr. &BFFE - 2*n*

(n = Anzahl der Parameter)

Die Parameter selbst werden ab Adresse &BFFE auf den Stapel gelegt (für jeden Parameter zwei Bytes). Danach kommen noch sechs weitere Bytes, die für die Simulation nicht von Bedeutung sind, auf den Stapel. Wird nun zum Beispiel ein Maschinenprogramm mit drei Parametern aufgerufen, so enthält das SP-Register (der Stackpointer, der immer auf den letzten Stapeleintrag zeigt) folglich die Adresse &BFF2. Ohne Parameter enthält das SP-Register die Adresse &BFF8. Dieser besondere Aufbau des Stapels und die speziellen Registerinhalte werden somit bei einem BASIC-Aufruf wirklichkeitsgetreu simuliert.

Aufruf von Maschinenebene

Zu Beginn der Simulation enthalten alle Register (bis auf PC und SP) den Wert Null. Das SP-Register hat stets den Inhalt &BFFE, da nur die Rücksprungadresse (symbolisiert durch den Wert &FFFF) auf dem Stapel liegt. Das Maschinenprogramm beendet die Simulation durch Ausgabe der *RETURN* -Meldung nur dann, wenn es auf einen RET-Befehl trifft und das SP-

Register den gleichen Wert enthält wie zu Beginn der Simulation (nur dann kann die korrekte Rücksprungadresse vom Stapel geholt werden).

Die Ausgabe auf dem Bildschirm erfolgt beim F-Register binär (die wichtigsten Flags sind beschriftet), beim a-, HL-, DE- und BC-Register dezimal und hexadezimal. Die übrigen Register (IX, IY, SP und PC) werden hexadezimal ausgegeben. Bei der dezimalen Ausgabe kann (bis auf das A-Register) eine vorzeichenfreie (0 bis 65535; dargestellt durch das + Symbol) oder eine vorzeichenbehaftete (-32768 bis +32767; dargestellt durch das +/- Symbol) Ausgabe gewählt werden. Dies kann während der Simulation jederzeit durch drücken der "/" -Taste erreicht werden. Ein Drücken der rechts davon (über der CTRL-Taste) liegenden Taste führt dazu, daß der untere Bildschirmteil, in dem die Ausgabe erfolgt, gelöscht und neu aufgebaut wird.

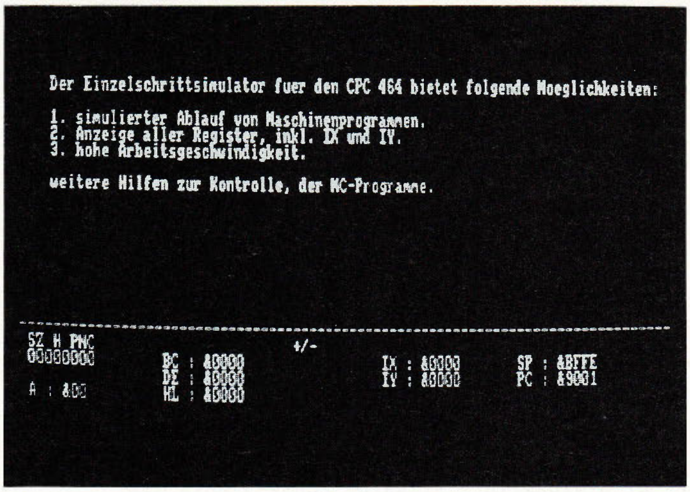
Durch Drücken der "." -Taste können Sie vom echten Einzelschrittmodus in eine fortlaufende Verarbeitung der Befehle (kein Warten auf einen Tastendruck nach jedem Befehl) umschalten. Durch erneutes Drücken dieser Taste gelangen Sie wieder in den ursprünglichen Modus. Bei der fortlaufenden Verarbeitung kann die Geschwindigkeit durch Drücken der "," -Taste in vier Stufen verändert werden. Ein Druck auf die ENTER-Taste führt, wie schon erwähnt, zum Abbruch der Simulation.

Das Funktionsprinzip

...des Einzelschrittsimulators läßt sich grob folgendermaßen beschreiben: nach der Einrichtung des simulierten Stapels werden die Register gespeichert. Bei einem Teil der Befehle (JP, CALL, RET, sämtliche Befehle, die das SP-Register beeinflussen etc.) werden diese vom Maschinenprogramm selbst manipuliert; mit den übrigen (LD, BIT, RES, SET etc.) wird jedoch wie folgt verfahren: ein solcher Befehl wird mit dem Opcode &C9 am Ende in einen bestimmten Speicherbereich kopiert, die Register werden mit den gespeicherten Werten geladen und der Befehl als normales Unterprogramm ausgeführt. Danach werden die dadurch veränderten Register und Flags wieder gespeichert und anschließend ausgegeben.

Bitte beachten Sie deswegen:

Solche tatsächlich ausgeführten Befehle, die das Maschinenprogramm ab Adresse &9C50 oder andere wichtige Adressen im RAM überschreiben, werden wahrscheinlich zum Absturz des Rechners führen. Dies gilt auch für Sprünge ins ROM mittels RST-Befehlen, die zum Absturz führen können (zum Beispiel ein Sprung nach &000). Die übrigen Sprünge ins ROM werden jedoch problemlos ausgeführt, wobei eine Ausgabe der Register jedoch erst wieder nach der Rückkehr ins RAM erfolgt. Beachten Sie außerdem, daß folgende Z80-Befehle nicht ausgeführt werden:



Der Einzelschrittsimulator, eine Hilfe für Assemblerfreunde

IM 0
IM 1
IM 2
RETI
RETN
sowie
EX AF,
'AF
EXX
RST &00
RST &38
HALT
DI
EI

Bei diesen Befehlen wird lediglich das PC-Register für den nächsten Befehl erhöht. Bei den letzten sieben Befehlen werden Sie zusätzlich noch akustisch darauf aufmerksam gemacht.

Das Abtippen

Listing 1 ist der BASIC-Teil des Programmes. Nach dem Abtippen sichern Sie ihn mit SAVE "Name auf Datenträger" (zum Beispiel: Diskette) ab. Listing 2 erzeugt nach dem Starten den Maschinencode und speichert ihn unter dem Namen eschmcode.BIN ab.

Der abgespeicherte Maschinencode wird dann vom Listing 1 nachgeladen und initialisiert. Das Listing 2 sollte aus Sicherheitsgründen vor dem Start abgespeichert werden. Danach ist es mit RUN zu starten.

(Manfred Doser/cd)

für 464

```
10 ' ***************************         [1051]
20 ' *                         *         [175]
30 ' *      Einzelschritt-     *         [1510]
40 ' *            Simulator    *         [1310]
50 ' *                         *         [175]
```

Listing EINZEL

```
60 ' *          1987 by        *          [648]
70 ' *       Manfred Doser     *          [538]
80 ' *                         *          [175]
90 ' ***************************          [1051]
100 '                                     [117]
110 MEMORY &1FFF                          [710]
120 LOAD "eschmcod.bin",&9C50             [1875]
130 DEF FNhigh(x)=INT(x/256):DEF FNlow(x)= [2550]
x-INT(x/256)*256
```

Listing EINZEL

```
140 MODE 2:INK 1,24:INK 0,1:BORDER 1:PEN 1 [3216]
:PAPER 0
150 PRINT "  EINZELSCHRITT-SIMULATOR":PRIN [4211]
T TAB(3)STRING$(28,"_")
160 FOR i=&9F56 TO &9F62:POKE i,0:NEXT i   [2383]
170 ix=42241                               [509]
180 CALL &9ED0                             [769]
190 LOCATE 1,7:INPUT "Startadresse : &",ad [2854]
r$
200 IF adr$="" THEN adr$="0"               [1181]
210 adr=VAL("&"+adr$):IF adr<0 THEN adr=ad [2014]
r+65536
220 LOCATE 1,10:INPUT "BASIC-Aufruf (j/n)  [2781]
? ",bm$
230 IF bm$<>"j" THEN bmfl=0:POKE &A2CC,0:P [7410]
OKE &A2CD,&C0:POKE &A2D9,0:POKE &A2DA,&C0:
GOTO 420:ELSE bmfl=1:POKE &9F59,&FF
240 LOCATE 1,13:PRINT "Anzahl der Paramete [2428]
r ?"
250 LOCATE 24,13:INPUT "",pz               [1618]
260 IF pz<0 OR pz>32 THEN LOCATE 24,13:PRI [3721]
NT STRING$(10," "):PRINT CHR$(7):GOTO 250
270 IF pz=0 THEN POKE &9F57,&68:POKE &9F5B [5561]
,FNlow(adr):POKE &9F5C,FNhigh(adr):ELSE PO
KE &9F57,&28
280 POKE &9F58,pz:POKE &9F5A,32-pz         [1173]
290 FOR i=1 TO pz                          [499]
300 LOCATE 17,17:PRINT STRING$(10," ")     [1628]
310 LOCATE 1,17:PRINT #0,USING "##";i;:PRI [3886]
NT ". Parameter : ";:INPUT "",pm
320 IF pm<0 OR pm>65536 THEN PRINT CHR$(7) [3113]
:GOTO 300
330 ix=ix+1:POKE ix,FNlow(pm):ix=ix+1:POKE [3272]
 ix,FNhigh(pm)
340 NEXT i                                 [375]
350 IF pz>0 THEN POKE &9F5B,FNlow(pm):POKE [2186]
 &9F5C,FNhigh(pm)
360 POKE &9F4E,FNlow(ix):POKE &9F4F,FNhigh [1985]
(ix)
370 ixrg=49150-pz*2                        [548]
380 POKE &9F5F,FNlow(ixrg):POKE &9F60,FNhi [2891]
gh(ixrg)
390 sp=49146-pz*2                          [1202]
400 POKE &A2CC,FNlow(sp):POKE &A2CD,FNhigh [1803]
(sp)
410 POKE &A2D9,FNlow(sp):POKE &A2DA,FNhigh [2483]
(sp)
420 MODE 2:POKE &9CFD,16:POKE &9CFE,1:POKE [3846]
 &9CFF,0
430 CALL &9C5D:CALL &9F6D,bmfl,adr         [1618]
440 MODE 2:PRINT "  EINZELSCHRITT-SIMULATO [3543]
R":PRINT TAB(3)STRING$(28,"_")
450 LOCATE 1,7:PRINT "Abbruch mit 'ENTER'" [3078]
460 CALL &9ED0:CALL &BB06:IF INKEY(18)=0 T [5374]
HEN MODE 2:PRINT " BASIC 1.0":PRINT:END
470 LOCATE 1,7:PRINT CHR$(20):GOTO 160     [1684]
```

```
10 ' BASIC-Lader zu "ESCHMCOD.BIN"             [2278]
20 '                                           [117]
30 ' am Ende jeder Zeile Pruefsumme            [2297]
40 '                                           [117]
100 DATA 00,00,01,00,01,00,00,00,00,81,69, [3080]
9C,00,21,50,9C,0295
110 DATA 11,01,00,01,01,00,C3,E9,BC,F3,21, [2463]
ED,9C,97,BE,28,0696
120 DATA 03,35,FB,C9,3E,16,CD,1E,BB,F3,28, [2083]
0A,21,FD,9C,CB,07A0
130 DATA 86,CB,E6,C3,E6,9C,3E,1E,CD,1E,BB, [2423]
F3,28,11,21,FD,08C8
140 DATA 9C,CB,E6,CB,4E,28,04,CB,8E,18,4B, [2784]
CB,CE,18,47,3E,0784
150 DATA 12,CD,1E,BB,F3,28,07,3E,02,32,56, [3478]
9F,18,38,3E,1F,04EE
160 DATA CD,1E,BB,F3,28,1E,21,FD,9C,CB,56, [3257]
28,13,CB,96,21,0777
170 DATA 3C,B5,36,15,23,AF,77,23,36,01,23, [2765]
77,23,77,18,16,0441
180 DATA CB,D6,18,12,3E,27,CD,1E,BB,F3,28, [2521]
0F,11,00,40,2A,057B
190 DATA FE,9C,19,22,FE,9C,3E,19,32,ED,9C, [2349]
FB,C9,00,00,00,0745
200 DATA 00,00,00,00,00,00,00,00,00,00,00, [1623]
00,00,05,01,00,0006
210 DATA 21,00,00,.39,01,0C,00,11,57,9F,ED, [2936]
B0,21,FD,9C,CB,0590
```

Listing EINZEL

```
220 DATA 46,20,3F,CB,C6,21,14,01,E5,CD,75, [3090]
BB,3E,14,CD,5A,06C7
230 DATA BB,E1,CD,75,BB,3E,2D,06,50,CD,5A, [1268]
BB,10,FB,DD,21,0845
240 DATA E2,9E,FD,21,0A,9F,06,09,C5,CD,88, [3912]
9E,C1,CD,96,9E,08D0
250 DATA 78,FE,09,28,0B,FD,E5,FD,21,21,9F, [3534]
CD,96,9E,FD,E1,0951
260 DATA 10,E6,21,FD,9C,CB,66,28,1B,CB,A6, [3425]
E5,21,15,22,CD,079F
270 DATA 75,BB,E1,CB,4E,28,06,FD,21,28,9F, [4048]
18,04,FD,21,25,069C
280 DATA 9F,CD,96,9E,DD,21,F4,9E,CD,88,9E, [2141]
3A,57,9F,4F,06,08A8
290 DATA 08,CB,21,C5,3E,30,30,01,3C,CD,5A, [2037]
BB,C1,10,F2,CD,0706
300 DATA 88,9E,3A,58,9F,F5,CD,A7,9E,CD,88, [2895]
9E,06,03,3E,20,07B8
310 DATA CD,5A,BB,10,FB,CD,88,9E,E1,6C,26, [3754]
00,CD,00,B9,CD,08A6
320 DATA 79,EE,06,05,21,62,9F,C5,E5,CD,88, [2968]
9E,E1,7E,2B,CD,0888
330 DATA A7,9E,7E,2B,CD,A7,9E,C1,10,ED,CD, [2964]
88,9E,2A,50,9F,08CA
340 DATA CD,BD,9E,CD,88,9E,2A,54,9F,CD,C2, [4336]
A2,CD,BD,9E,06,0997
350 DATA 03,21,15,21,E5,E1,2C,E5,C5,CD,75, [3392]
BB,06,06,3E,20,065D
360 DATA CD,5A,BB,10,FB,C1,10,ED,E1,3A,FD, [2615]
9C,CB,4F,28,05,08A6
370 DATA 21,79,EE,18,03,21,C5,9E,22,1E,9E, [3649]
22,2A,9E,22,36,0547
380 DATA 9E,CD,00,B9,21,16,21,CD,75,BB,2A, [3463]
59,9F,CD,C5,9E,07CB
390 DATA 21,17,21,CD,75,BB,2A,5B,9F,CD,C5, [3140]
9E,21,18,21,CD,06D1
400 DATA 75,BB,2A,5D,9F,CD,C5,9E,3A,56,9F, [3704]
FE,02,28,3A,FE,0815
410 DATA 01,20,15,21,19,30,CD,75,BB,FD,21, [3216]
2B,9F,CD,96,9E,0686
420 DATA CD,D0,9E,CD,06,BB,18,21,3A,FD,9C, [2557]
CB,57,28,13,2A,075C
430 DATA FE,9C,2B,7C,B5,FE,00,20,F9,3A,56, [2592]
9F,FE,02,28,09,076D
440 DATA 18,13,CD,06,BB,FE,0D,20,0C,E1,E1, [3253]
E1,E1,E1,E1,21,0857
450 DATA 50,9C,C3,EC,BC,C3,E1,9F,DD,6E,00, [2809]
DD,23,DD,66,00,0928
460 DATA DD,23,CD,75,BB,C9,FD,7E,00,FD,23, [2115]
CB,7F,CB,BF,C2,09F7
470 DATA 5A,BB,CD,5A,BB,18,EF,F5,1F,1F,1F, [3008]
1F,CD,B0,9E,F1,087B
480 DATA E6,0F,FE,0A,38,02,C6,07,C6,30,C3, [3109]
5A,BB,7C,CD,A7,07C2
490 DATA 9E,7D,C3,A7,9E,3E,02,32,C1,B0,22, [2594]
C2,B0,C3,36,F2,0885
500 DATA F3,21,3C,B5,36,15,23,AF,77,23,36, [3102]
01,23,77,23,77,0527
510 DATA FB,C9,15,02,18,02,16,12,17,12,18, [2503]
12,16,2D,17,2D,02F7
520 DATA 16,3E,17,3E,16,02,18,07,18,0B,18, [3266]
0B,17,33,16,33,01B9
530 DATA 18,18,17,18,16,18,17,44,16,44,53, [3487]
5A,20,48,20,50,02C7
540 DATA 4E,C3,C1,42,C3,44,C5,48,CC,49,D8, [2933]
49,D9,53,D0,50,08AA
550 DATA C3,20,3A,20,A6,2B,2F,AD,20,2B,A0, [2481]
2A,20,52,45,54,050A
560 DATA 55,52,4E,20,AA,00,00,00,00,00,00, [2579]
00,00,00,00,00,01BF
570 DATA 00,00,00,00,00,00,00,00,00,00,00, [1562]
00,00,00,01,A5,00A6
580 DATA FF,FF,00,A5,00,A5,01,A2,F4,00,00, [2131]
00,00,00,00,00,04DF
590 DATA 00,00,00,EA,F1,86,7F,9A,B9,86,7F, [1996]
EA,F1,ED,53,50,08A3
600 DATA 9F,DD,7E,02,FE,01,20,4A,21,8B,DD, [2422]
22,00,A5,ED,5B,06FD
610 DATA 4E,9F,D5,13,21,63,9F,01,06,00,ED, [3175]
B0,EB,22,54,9F,069C
620 DATA 2B,2B,22,52,9F,E1,11,01,A5,B7,ED, [2426]
52,28,17,45,CB,0646
630 DATA 38,0E,FF,11,FD,BF,21,02,A5,1B,ED, [2002]
A0,ED,A0,1B,1B,0745
640 DATA 1B,10,F6,18,03,11,FD,BF,21,6C,9F, [3337]
01,06,00,ED,B8,05E1
650 DATA 18,14,21,FF,FF,22,00,A5,21,00,A5, [2880]
22,52,9F,23,23,0531
```

Listing EINZEL

```
660 DATA 22,54,9F,11,F3,BF,21,62,9F,01,0C, [3583]
00,ED,B8,13,EB,06AA
670 DATA F9,2A,50,9F,7E,16,00,FE,C9,CA,7C, [2727]
A0,14,14,FE,C3,083C
680 DATA CA,7C,A0,14,14,FE,CD,CA,7C,A0,5F, [3065]
16,01,0E,08,06,0651
690 DATA 06,3E,10,BB,28,1A,81,10,FA,16,00, [2332]
3E,C0,82,06,08,0480
700 DATA BB,28,15,81,10,FA,14,14,7A,FE,06, [2783]
20,EE,C3,02,A1,069D
710 DATA CB,20,3E,08,80,47,18,02,CB,20,21, [2162]
EC,A0,48,06,00,04F8
720 DATA 09,D5,5E,23,56,EB,D1,F1,F5,E9,FA, [2493]
7C,A0,18,28,F2,0988
730 DATA 7C,A0,18,23,EA,7C,A0,18,1E,E2,7C, [2099]
A0,18,19,38,2C,0626
740 DATA 18,15,30,28,18,11,28,24,18,0D,20, [2459]
20,18,09,18,1C,01B4
750 DATA F1,C1,05,C5,F5,20,15,7A,3C,FE,03, [2168]
38,02,3E,03,47,061F
760 DATA 2A,50,9F,23,10,FD,22,50,9F,C3,00, [2765]
9D,7A,FE,01,28,065B
770 DATA 35,FE,00,28,18,2A,50,9F,23,5E,23, [2495]
56,ED,53,50,9F,05B5
780 DATA FE,02,CA,00,9D,23,EB,CD,D6,A0,C3, [1855]
00,9D,CD,E1,A0,0966
790 DATA ED,53,50,9F,ED,5B,52,9F,B7,ED,52, [3774]
C2,00,9D,3E,01,07FC
800 DATA 32,56,9F,C3,00,9D,2A,50,9F,23,5E, [2680]
23,16,00,CB,7B,05A0
810 DATA 28,0D,7B,2F,3C,5F,B7,ED,52,22,50, [2119]
9F,C3,00,9D,19,05FA
820 DATA 22,50,9F,C3,00,9D,2A,54,9F,73,23, [2330]
72,23,22,54,9F,05CE
830 DATA C9,2A,54,9F,2B,56,2B,5E,22,54,9F, [3305]
C9,00,00,3A,A0,05A8
840 DATA 3F,A0,44,A0,49,A0,4E,A0,52,A0,56, [3922]
A0,5A,A0,5E,A0,077A
850 DATA 60,A0,0E,10,3E,C5,06,04,BB,CA,1C, [2436]
A1,81,10,F9,3E,0635
860 DATA C1,06,04,BB,28,06,81,10,FA,C3,6B, [2895]
A1,2A,50,9F,23,064A
870 DATA 22,50,9F,CB,57,28,21,78,FE,01,20, [3483]
0A,ED,5B,57,9F,065B
880 DATA CD,D6,A0,C3,00,9D,05,CB,20,21,5F, [3364]
9F,2B,10,FD,5E,0748
890 DATA 23,56,CD,D6,A0,C3,00,9D,CD,E1,A0, [2605]
78,FE,01,20,05,0806
900 DATA E1,D5,C3,00,9D,05,3E,31,91,10,FD, [2924]
21,F0,A4,77,ED,0841
910 DATA 53,F1,A4,3E,C9,32,F3,A4,C3,10,A3, [1420]
3E,E3,BB,C2,93,095F
920 DATA A1,CD,E1,A0,3E,21,32,F0,A4,ED,53, [2133]
F1,A4,3E,C9,32,0922
930 DATA F3,A4,ED,5B,5D,9F,CD,D6,A0,2A,50, [2877]
9F,23,22,50,9F,086B
940 DATA C3,10,A3,3E,E9,BB,20,09,2A,5D,9F, [3871]
22,50,9F,C3,00,067B
950 DATA 9D,3E,F9,BB,20,13,2A,50,9F,23,22, [2945]
50,9F,2A,5D,9F,0635
960 DATA CD,C2,A2,22,54,9F,C3,00,9D,3E,39, [2679]
BB,20,22,2A,54,0698
970 DATA 9F,CD,C2,A2,ED,5B,5D,9F,19,3E,21, [1771]
32,F0,A4,22,F1,0865
980 DATA A4,2A,F3,A4,36,C9,2A,50,9F,23,22, [2919]
50,9F,C3,10,A3,0727
990 DATA 2A,50,9F,23,3E,FD,BB,28,06,D6,20, [3090]
BB,C2,26,A3,16,06B2
1000 DATA 04,47,7E,FE,36,CA,FB,A2,FE,21,CA [2643]
,FB,A2,FE,2A,CA,09DC
1010 DATA FB,A2,FE,22,CA,FB,A2,FE,CB,CA,FB [2460]
,A2,FE,E3,20,17,0B6C
1020 DATA CD,DD,A2,FE,DD,28,06,ED,5B,61,9F [2234]
,18,04,ED,5B,5F,0860
1030 DATA 9F,CD,D6,A0,C3,10,A3,FE,F9,20,19 [3452]
,78,FE,DD,28,05,0908
1040 DATA 2A,61,9F,18,03,2A,5F,9F,CD,C2,A2 [3023]
,22,54,9F,CD,F2,0772
1050 DATA A2,C3,00,9D,FE,E5,20,18,78,FE,DD [2698]
,28,06,ED,5B,61,0847
1060 DATA 9F,18,04,ED,5B,5F,9F,CD,D6,A0,CD [2027]
,F2,A2,C3,00,9D,0905
1070 DATA FE,E1,20,06,CD,DD,A2,C3,10,A3,16 [2059]
,02,FE,09,CA,FB,08AB
1080 DATA A2,FE,19,CA,FB,A2,FE,29,CA,FB,A2 [2386]
,FE,23,CA,FB,A2,0B36
1090 DATA FE,2B,CA,FB,A2,FE,E9,20,13,78,FE [2753]
,DD,28,05,2A,61,08B5
```

Listing EINZEL

```
1100 DATA 9F,18,03,2A,5F,9F,22,50,9F,C3,00 [2069]
,9D,FE,39,20,1D,05C7
1110 DATA 2A,54,9F,CD,C2,A2,78,FE,DD,28,06 [3305]
,ED,5B,61,9F,18,082F
1120 DATA 04,ED,5B,5F,9F,19,EB,CD,E0,A2,C3 [3538]
,10,A3,16,03,C3,07EF
1130 DATA FB,A2,ED,5B,52,9F,B7,ED,52,38,08 [2445]
,11,00,C0,EB,B7,087F
1140 DATA ED,52,C9,19,EB,B7,ED,52,11,00,C0 [2539]
,19,C9,CD,E1,A0,0903
1150 DATA 3E,21,32,F1,A4,78,32,F0,A4,ED,53 [3526]
,F2,A4,21,F4,A4,08F3
1160 DATA 36,C9,2A,50,9F,23,23,22,50,9F,C9 [2978]
,42,2A,50,9F,E5,0678
1170 DATA 23,10,FD,22,50,9F,4A,E1,11,F0,A4 [3819]
,ED,B0,EB,36,C9,0898
1180 DATA F1,C1,D1,E1,DD,E1,FD,E1,CD,F0,A4 [2437]
,FD,E5,DD,E5,E5,0DEA
1190 DATA D5,C5,F5,C3,00,9D,3E,ED,BB,C2,FD [2858]
,A3,2A,50,9F,23,0973
1200 DATA 7E,FE,73,20,18,23,5E,23,56,23,22 [3723]
,50,9F,ED,53,48,05DD
1210 DATA A3,2A,54,9F,CD,C2,A2,22,00,00,C3 [1385]
,00,9D,21,97,A3,06CE
1220 DATA 06,04,BE,28,05,23,10,FA,18,41,D6 [2544]
,10,32,80,A3,CB,0581
1230 DATA 47,20,05,21,00,00,18,03,21,28,03 [1808]
,22,89,A3,CD,F2,0401
1240 DATA A2,21,77,A3,22,86,9E,F1,C1,D1,E1 [2858]
,DD,E1,FD,E1,ED,0B10
1250 DATA 00,FD,E5,DD,E5,E5,D5,C5,F5,00,00 [2630]
,EA,00,9D,21,E1,09A1
1260 DATA 9F,22,86,9E,C3,00,9D,B0,B8,B1,B9 [2818]
,21,AD,A3,06,04,0792
1270 DATA BE,28,05,23,10,FA,18,09,16,04,C3 [1968]
,FB,A2,4B,5B,43,059C
1280 DATA 53,FE,7A,28,04,FE,72,20,26,D6,20 [2726]
,32,CB,A3,2A,54,06C1
1290 DATA 9F,CD,C2,A2,EB,2A,5D,9F,F1,F5,ED [2287]
,00,3E,21,32,F0,0935
1300 DATA A4,22,F1,A4,3E,C9,32,F3,A4,CD,F2 [2233]
,A2,C3,10,A3,21,0923
1310 DATA F8,A3,06,05,BE,28,08,23,10,FA,16 [2958]
,02,C3,FB,A2,CD,0706
1320 DATA F2,A2,00,00,00,C3,00,9D,46,56,5E [1347]
,4D,45,21,4F,A4,0594
1330 DATA 06,04,7E,BB,28,05,23,10,F9,18,48 [2712]
,F5,78,FE,03,38,05A2
1340 DATA 1F,CD,E1,A0,2A,50,9F,E5,ED,53,50 [3716]
,9F,2A,54,9F,ED,08A4
1350 DATA 5B,52,9F,B7,ED,52,20,05,3E,01,32 [2464]
,56,9F,E1,18,0B,05D1
1360 DATA 2A,50,9F,E5,23,23,23,22,50,9F,E1 [3436]
,F1,32,F0,A4,23,0733
1370 DATA 5E,23,56,ED,53,F1,A4,3E,C9,32,F3 [2677]
,A4,C3,10,A3,CF,08C1
1380 DATA EF,D7,DF,21,66,A4,06,07,7E,BB,28 [2818]
,05,23,10,F9,18,0687
1390 DATA 0C,16,03,C3,FB,A2,32,3A,01,11,21 [2701]
,2A,22,21,8A,A4,04BF
1400 DATA 06,07,7E,BB,28,05,23,10,F9,18,16 [2948]
,2A,50,9F,23,22,042B
1410 DATA 50,9F,3E,07,CD,5A,BB,C3,00,9D,08 [2898]
,D9,C7,FF,76,F3,0886
1420 DATA FB,3E,33,BB,20,06,2A,54,9F,23,18 [3085]
,09,3E,3B,BB,20,0502
1430 DATA 11,2A,54,9F,2B,22,54,9F,2A,50,9F [1634]
,23,22,50,9F,C3,057E
1440 DATA 00,9D,3E,06,16,02,06,08,0E,08,BB [4169]
,CA,FB,A2,81,10,04D0
1450 DATA F9,3E,C6,06,08,BB,CA,FB,A2,81,10 [3190]
,F9,3E,CB,BB,CA,0945
1460 DATA FB,A2,16,01,C3,FB,A2,00,00,00,00 [3055]
,00,00,00,00,00,0414
1470 zeile=1:adr=40016                      [1823]
1480 FOR loop1%=1 TO 137                    [748]
1490 summe=0:FOR loop2%=1 TO 16             [1098]
1500 READ byte$:byte=VAL("&"+byte$):POKE a [2844]
dr,byte
1510 summe=summe+byte:adr=adr+1:NEXT loop2 [3487]
%
1520 READ pruefsum$:pruefsum=VAL("&"+pruef [2374]
sum$)
1530 IF pruefsum<>summe THEN PRINT"Fehler  [3738]
in Zeile :";zeile:END
1540 zeile=zeile+1:NEXT loop1%              [1150]
1550 SAVE"eschmcod.bin",b,40016 ,2183       [1612]
```

Listing EINZEL

CPC-Fitneß-Center

Joystickmatte im Eigenbau

Heute zeigt sich der CPC mal ausgesprochen sportlich. Eine Bauanleitung für eine Joystickmatte mit der passenden Software verwandelt Ihren CPC in ein kleines Fitneß-Center. Ob nun als Partygag oder als Entspannungsmöglichkeit, wenn man zu lange am Computer gearbeitet hat, bietet diese Matte neue Möglichkeiten.

Das Programm

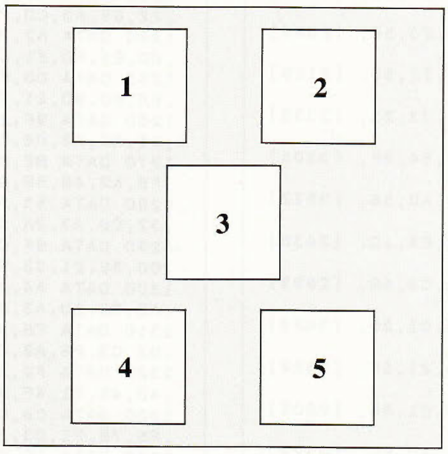
Das Programm bietet acht verschiedene Übungen an:

- **Jogging**
- **Rumpfbeugen**
- **Liegestütze**
- **Springen**
- **Sit-up**
- **Kniebeugen**
- **Liegestützwechselhüpfen**
- **Entspannen**

Es können bis zu neun Spieler an diesem Wettkampf teilnehmen. Drei Schwierigkeitsstufen stehen zur Auswahl. Die Highscore-Liste kann abgespeichert und auf Wunsch geladen werden. Falls man sich in seiner Gesamtpunktzahl verbessert, wird die Liste entsprechend aktualisiert. Das Programm wurde großzügig mit mehrstimmiger Musik versehen. Zu erwähnen ist hier noch, daß von der Möglichkeit Gebrauch gemacht wurde, Musikstücke auch in BASIC unter Interrupt zu spielen. Bei der Grafik mußte ein Kompromiß in Kauf genommen werden. Zum einen sollen die Sprites möglichst groß sein, damit man sie von der Matte aus gut erkennen kann, zum anderen möchte man möglichst wenig abtippen müssen. Aus diesem Grund wurden 22 kleine Sprites entworfen und dann vergrößert.

Das Spiel und die Matte

Die Matte:



Die Matte wird einfach an den Joystickport angeschlossen. Die numerierten Felder sind Kontaktplatten, die wie Schalter eines Joysticks auf Druck reagieren.

1. Jogging
Man stellt sich auf die Flächen 4 und 5 und tritt abwechselnd auf diese Flächen. Je schneller man dies macht, desto höher wird die Punktzahl.

2. Rumpfbeugen
Mit den Füßen stellt man sich auf die Flächen 1 und 2 und bewegt die Arme bei durchgestreckten Beinen auf die Fläche 3. Dies so oft wie möglich wiederholen. Die Füße müssen immer auf den Flächen 1 und 2 stehenbleiben.

3. Liegestütze
Die Hände werden auf die Platten 1 und 2 gelegt, dann wird der Körper immer so weit nach unten bewegt, bis man mit dem Oberkörper die Fläche 3 berührt. Auch hier gilt es, möglichst viele Liegestützen in möglichst kurzer Zeit zu schaffen.

4. Springen
Die Beine werden auf die Platten 1 und 2 gestellt, dann wird mit beiden Beinen auf die Fläche 3 gesprungen, dann wieder zu den Flächen 1 und 2 zurück und so weiter.

5. Sit-up
Man legt sich so hin, daß die Füße auf den Flächen 1 und 2 liegen. Dann wird der Oberkörper mit den Armen zur Fläche 3 geführt und wieder zurück.

6. Kniebeugen
Mit den Füßen stellt man sich auf die Flächen 4 und 5, dann geht man in die Hocke und berührt mit den Händen die Fläche 3. Nun streckt man sich wieder und beginnt von vorne.

7. Liegestützwechselhüpfen
Man nimmt die Haltung wie bei den Liegestützen ein. Die Arme bleiben gestreckt. Die Beine werden jedoch immer angezogen und gestreckt. Wenn sie angezogen sind, müssen sich die Füße auf den Flächen 4 und 5 befinden.

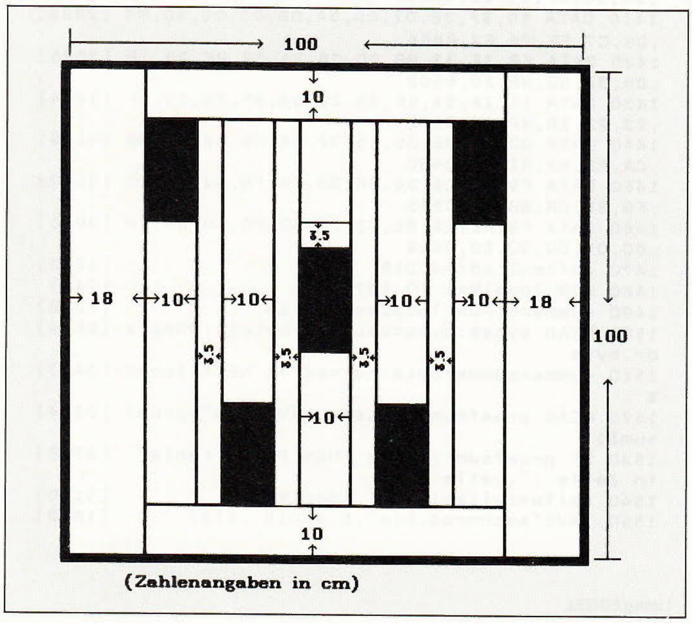
8. Entspannen
Beliebige Felder werden mit den Füßen berührt. Es darf jedoch nicht ein Feld gedrückt gehalten werden, während man ein anderes berührt. Um am Ende einer Übung in die nächste zu gelangen, wird ausschließlich das Feld 3 berührt (entspricht 'Fire').

So, das waren genug trockene Erklärungen, am besten ausprobieren! Zu Beginn des Programms wird die Anzahl der Spieler eingegeben. Wenn hier '0' eingegeben wird, geht das Programm davon aus, daß die Highscore-Daten und Namen der Spieler von der Diskette geholt werden sollen und fragt nach dem Filenamen. Wird kein entsprechendes File gefunden, dann wird das Inhaltsverzeichnis ausgegeben und das Programm neu gestartet. Wenn man keine Daten laden möchte, werden nun die Namen der Mitspieler eingegeben. Schließlich kann man noch den Schwierigkeitsgrad auswählen. Kassettenbenutzer sollten die Listings in folgender Reihenfolge abtippen:
1.Listing: *FITSP.BAS*, erzeugt *FITSP.BIN* (Sprites)
2.Listing: *FITMC.BAS*, erzeugt *FITMC.BIN* (MC-Teil)
3.Listing: *FITNESS.BAS* (Hauptprogramm), lädt *SPM.BIN* und *FITMC.BIN* nach.
Für Diskettenbenutzer spielt die Reihenfolge keine Rolle.

Was man für die Matte benötigt:

Was die Zahlen in Klammern bedeuten, ist den Bezugsquellenhinweisen zu entnehmen.

2 x Baufolie, transparent 1m x 1m ca. 0.14 mm dick (1)
2 x Gewebeklebeband 10m/50mm breit (2,3)
3 x Gummiringe für Einmachgläser (2)
1 x Joystickstecker, eventuell mit Kabel (4,5)
10 x Pertinax-Platten (Platinen) 1,5 mm stark mit einseitiger Kupferauflage 160mm x 100 mm (4,5)
5 m Mikrofonkabel 5polig + Abschirmung (4,5)



Skizze 1: Aufteilungsschema der Folie

1 x Kabelring-Litze (10m) Querschnitt ca. 0,14 mm (4,5)
1 x Rolle Isolierband
– Hefter mit Heftklammern
– Lötkolben mit etwas Lötzinn
– Lineal, Schere (Messer, Rasierklinge), Folienstift (wasserlöslich)
– Schleifpapier

Bezugsquellenhinweise

(1) Baugeschäft, Geschäfte für den Dachdeckerbedarf, Baumärkte, teilweise auch Holzhandlungen
(2) Supermarkt, Warenhaus
(3) Bastel-Laden
(4) Elektronik-Laden
(5) Elektronik-Versand, z.B Conrad Elektronik, Klaus-Conrad-Str. 1, 8452 Hirschau.

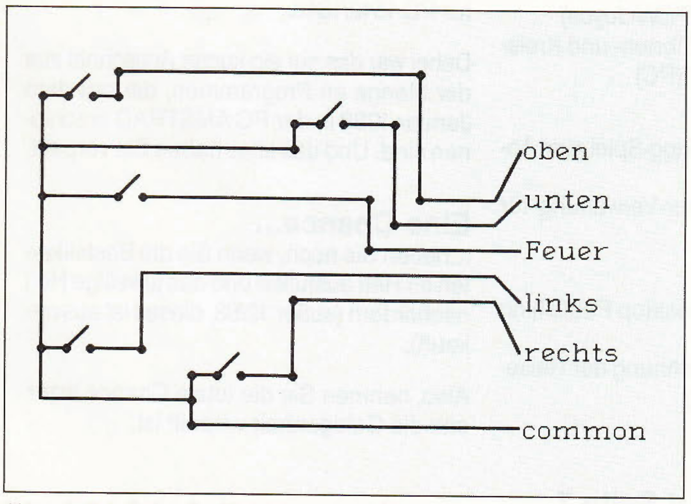
Die Maße der angegebenen Bauteile sind allerdings nicht bindend, denn beispielsweise wäre statt Folie auch Stoff denkbar, und ob die Litze etwas dünner oder dicker ist, spielt keine besonders große Rolle.

Wie wird's nun gemacht?

1. Schneiden Sie die zwei Folienstücke auf 1 m x 1 m zu.
2. Auf eine dieser Folien zeichnen Sie nun das Aufteilungsschema.
3. Mit etwas Schleifpapier runden Sie die Kanten und Ecken der Platinen etwas ab (vor allem die Kanten, auf denen sich kein Kupfer befindet). Dies dient zum Schutz der Folie.
4. Von den Gummiringen schneiden Sie nun 20 Stücke à 1,5 cm ab und kleben jeweils vier davon auf die Ecken der Kupferseite der Platinen (das heißt auf jede Ecke ein Gummistück). Fünf Platinen werden so präpariert.
5. Von Ihrem Litzen-Kabelring schneiden Sie vier Kabelstücke à 60 cm ab.
6. Die präparierten Platinen werden auf die schraffierten Flächen der Folie gelegt (siehe Zeichnung) und mit Hilfe eines Lötkolbens und den Kabelstücken untereinander verbunden. Die Kabel sollten direkt neben den Gummistücken auf den Platinen angelötet werden.
7. Nun wird das Mikrofonkabel auf 90 cm abisoliert (nur die äußere Hülle) und die Kabel aus der Abschirmung (= die nicht isolierte Litze) herausgezogen, so daß man am Ende sechs einzelne Leitungen erhält.
8. Neben den bereits verbundenen Platinen wird jeweils eine andere Platine gelegt. Die Abschirmung wird mit der präparierten Platte rechts unten verbunden, sie dient als gemeinsame Masse-Leitung. An die danebenliegenden, nicht präparierten Platinen werden die anderen Leitungen des Mikrofonkabels angelötet (die Farbe der Kabel spielt für die Anordnung keine Rolle). Es muß nun darauf geachtet werden, daß beim Übereinanderlegen einer präparierten Platte und der dazugehörigen Platine nicht zwei Lötstellen übereinanderliegen, das heißt, solange kein Druck auf die Platinen ausgeübt wird, darf zwischen den Platinen keine leitende Verbindung entstehen.
9. Wenn zwei solcher Platten übereinander gelegt werden, dann halten die Gummistücke die Kupferflächen auseinander. Sobald die Platten jedoch zusammengedrückt werden, schließt sich der Stromkreis.
10. Nun werden jeweils zwei zusammengehörige Platten mit den Kupferflächen nach innen übereinander gelegt und mit Isolierband so verbunden, daß sie nicht verrutschen können. Die Kabelverbindungen bleiben so lang, damit man die Matte später falten kann.
11. Jetzt wird das Joystickkabel mit dem Mikrofonkabel verbunden. Die Abschirmung des Mikrofonkabels wird mit der schwarzen Leitung des Joystickkabels verbunden. Wenn Sie einen Joystickstecker verwenden, dann wird der 'Common'-Anschluß (aus dem CPC-Bedienungshandbuch zu entnehmen) mit der Abschirmung verbunden. Da die Platinen noch nicht verdeckt sind, kann man sehen, welche Mikrofonleitungen zu welcher Platte gehen. Drücken Sie nun die mittlere Kontaktfläche, und berühren Sie mit der zu dieser Platte führenden Leitung die einzelnen Leitungen des Joystickkabels, bis ein 'X' = Fire auf dem Bildschirm erscheint. Diese Kabel gehören zusammen. Entsprechend wird mit den anderen Flächen verfahren, nur müssen dort andere Zeichen erscheinen: **Pfeil nach unten = unten, Pfeil nach oben = oben, X = Fire, Pfeil nach links = links, Pfeil nach rechts = rechts.**
12. Sind die entsprechenden Verbindungen hergestellt, wird das zweite Stück Folie über die Platinen gelegt, die bereits auf der anderen Folie liegen. Das Mikrofonkabel wird seitlich herausgeführt. Man kann es mit einem Knoten versehen, der als Zugentlastung dient. Der Knoten muß dann noch von der Folie verdeckt sein.
13. Um die Kontaktflächen herum werden Heftklammern gesetzt, die verhindern, daß die Platten verrutschen. Es empfiehlt sich, die Matte auf einen weichen Untergrund zu legen, um die Heftklammern leicht durch die Folie drücken zu können. Dann wird die Matte gewendet und die hervorstehenden Enden der Heftklammern umgebogen, dafür sollte besser ein fester Untergrund vorliegen.
14. Sind auf diese Weise alle Platinen befestigt, wird der Rand der Matte um 2 cm umgeknickt und mit dem Hefter und Klammern zusammengeheftet.
15. Mit dem Gewebe-Klebeband werden die Kontaktflächen und die Ränder der Matte beklebt (von beiden Seiten).

Alle Heftklammern müssen vom Klebeband bedeckt sein. Nun wird die Matte einfach an den Joystickport angeschlossen und ist einsatzbereit.

(Klaus Eberle/jb)



Skizze 2: Schaltplan für die einzelnen Leitungen mit der jeweiligen Richtungsangabe

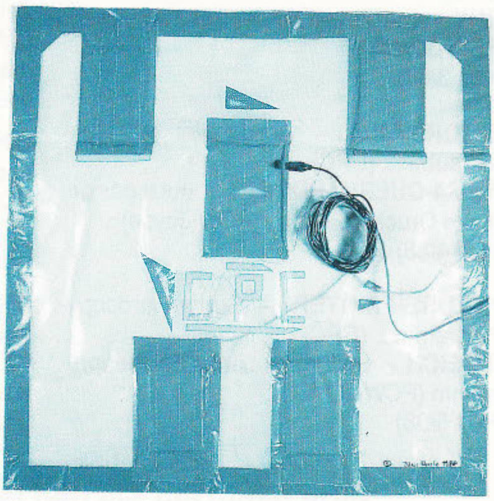


Bild 1. Das ist die komplette Joystickmatte im Eigenbau



Wissen Sie eigentlich, was Sie versäumt haben?

Nicht nur, daß Ihnen eine geballte Fülle an Informationen fehlt, Sie haben gleichzeitig jeden Monat das Superprogramm für Ihren **CPC**, **PCW** oder **PC** verpaßt. Entgangen sind Ihnen höchstwahrscheinlich seit Januar 1988:

POPCORN – das Super-Strategiespiel... (CPC)
LOCOCON – schnelle Konvertierung von LocoScript- Texten... (PCW-Joyce)
(Heft 1/88)

DESKMAN – Komfortable Benutzeroberfläche für Diskettenoperationen... (CPC)
SCREENY – Grafik-Module ohne GSX für Mallard- BASIC... (PCW-Joyce)
(Heft 2/88)

SCHREIBMASCHINENTRAINER – zum Erlernen der Zehnfinger-Schreibweise... (CPC)
TINY – Ein Texteditor der Sonderklasse... (PCW-Joyce)
(Heft 3/88)

ROTORMANIA – Actionspiel mit Hubschrauber... (CPC)
DIN-A4-QUERHARDCOPY – nutzt das gesamte Druckerpapier... (PCW-Joyce)
(Heft 4/88)

PICTURE-PRINTER – Super-Hardcopy-Programm... (CPC)
3DZEICH – vektororientiertes Zeichenprogramm (PCW-JOYCE)
(Heft 5/88)

CPC-ASSEMBLER V 2.0 – Der Z80-Assembler zum Abtippen... (CPC)
XX-FORMAT – 188 kByte freie Kapazität auf Diskette... (PCW-Joyce)
FONTEDIT – Neue Zeichensätze unter BASIC2... (PC)
(Heft 6/88)

MAGIC SCREEN – Manipulation von Grafiken... (CPC)
MONITORVERBESSERUNG – Klares Bild auf CGA-Monitoren... (PC)
(Heft 7/88)

BACKGAMMON – DIE Super-Simulation des Brettspiels... (CPC)
MILLION – BASIC2-Spiel ums liebe Geld... (PC)
(Heft 8/88)

RSX-SYMBOL-DESIGNER – Zeicheneditor der Spitzenklasse... (CPC)
3D-GRAFIK – Fortsetzung des 3D-Zeichenprogrammes... (PCW-Joyce)
DIAGRAMM – Balken-, Torten- und Kreisdiagramme in BASIC2... (PC)
(Heft 9/88)

LOOK – Das tolle Mahjongg-Spiel zum Abtippen... (CPC)
SCHOOLDAT – Literatur-Verwaltung für PC 1512/1640... (PC)
(Heft 10/88)

ARTWORX – Riesig: Desktop Publishing auf dem CPC... (CPC)
REISEKOSTEN – Abrechnung der Reisekosten in BASIC2... (PC)
(Heft 11/88)

Nicht zu vergessen die Highlights dieses Jahres:

MAZE-GLIDER (Spiel, 1/89),
ANIMATOR (Anwendung, 2/89),
TEXT-EDIT DE LUXE (Anw., 3/89),
SOUNDMANAGER (Anw., 4/89),
AUSTRALIEN (Spiel, 5/89),
VIDEODATEI (Anw., 6/89),
für den CPC.

FILE RESCUE (Anw., 1/89),
BACKGAMMON (Spiel, 2/89),
MINI-LEXIKON (Anw., 4/89),
KASSETTENLABEL (Anw.,5/89),
für PCW-Joyce

FARBE (Anw., 1/89),
KONTOFÜHRUNG (Anw., 2/89),
VOKABELTRAINER (Anw., 3/89),
BÖRSENSIMULATION (Spiel, 5/89),
REGENT (Spiel, 6/89),
für PC 1512/1640.

Dabei war das nur ein kurzer Ausschnitt aus der Menge an Programmen, die seit dem Januar 1988 in der PC AMSTRAD erschienen sind. Und das alles haben Sie verpaßt.

Eine Chance...

...haben Sie noch, wenn Sie die Bestellkarten im Heft ausfüllen und das jeweilige Heft nachordern (außer 12/88, dieses ist ausverkauft).

Also, nehmen Sie die letzte Chance wahr, ehe die Gelegenheit verpaßt ist.

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

Für 464-664-6128

```
1000 '==== Fitness-Center ====                [2203]
1010 KEY 138,"mode 2:ink 0,13:ink 1,0:pen    [8548]
1:paper 0:list ":MEMORY 26999:INK 0,15:BOR
DER 15:PEN 15:MODE 0:GOTO 1090
1020 FOR i=25 TO 0 STEP -1:OUT &BC00,6:OUT   [7702]
 &BD00,i:CALL &BD19:CALL &BD19:NEXT:FOR i=
0 TO 15:INK i,15:NEXT:RETURN
1030 n=0:fl=1:e=0:¦FBLOCK,adr(n):n=n+1:IF   [4022]
jump THEN ¦SET,5:¦FBLOCK,33912:¦SET,0
1040 GOSUB 4400:GOSUB 1250:GOSUB 1740:EVER  [5317]
Y 50,0 GOSUB 1470:RETURN
1050 ¦INIT:IF gfl<>1 THEN FOR i=0 TO 2:¦SE  [7180]
T,0:¦WINDOW,x,y,xb,yb:¦FBLOCK,sp(i):¦XZOOM
,4:¦YZOOM,5
1060 IF gfl=2 THEN ¦WINDOW,0,0,24,96        [1802]
1070 IF gfl<>1 THEN ¦GETSPR,adr(i):NEXT:RE  [1924]
TURN
1080 IF gfl=1 THEN FOR i=0 TO 1:¦SET,0:¦WI  [12639]
NDOW,x,y,xb,yb:¦FBLOCK,sp(i):¦XZOOM,4:¦YZO
OM,5:¦GETSPR,adr(i):NEXT:¦SET,0:¦WINDOW,0,
0,xb,yb:¦FBLOCK,sp(2):¦XZOOM,4:¦YZOOM,5:¦G
ETSPR,adr(3):RETURN
1090 GOSUB 1020                              [863]
1100 DEFINT a-z                              [553]
1110 DIM erg(40),adr(4),punkte(9),alt(9),s  [2601]
p(2)
1120 sc=0:spieler=1:MODE 0                   [732]
1130 POKE &A610,&3E:POKE &A611,0:POKE &A61  [5373]
2,&C3:POKE &A613,&E4:POKE &A614,&BB:CALL &
A600
1140 LOAD"fitsp.bin",&9585                  [2160]
1150 LOAD"fitmc.bin",40000:CALL 40000:¦INI  [3465]
T:¦SET,0
1160 GOSUB 3510:GOSUB 1020:MODE 0:power=&9  [2695]
037+2000
1170 FOR art=1 TO 8:FOR num=0 TO spieler-1  [9685]
:E=0:ON art GOSUB 1510,1900,2080,2210,2560
,2700,2840,2980:GOSUB 1290:punkte(num)=pun
kte(num)+sc:GOSUB 1020:NEXT:NEXT
1180 FOR i=0 TO 8                            [489]
1190 IF alt(i)>punkte(i) THEN punkte(i)=al  [3743]
t(i)
1200 NEXT                                    [350]
1210 MODE 0:GOSUB 4040:CLEAR:GOTO 1160      [2507]
1220 '                                       [117]
1230 'INK                                    [151]
1240 '                                       [117]
1250 OUT &BC00,6:OUT &BD00,0:INK 0,11:INK   [11044]
1,5:INK 2,24:INK 3,19:INK 4,26:INK 5,0:INK
 6,23:INK 7,1:INK 8,3:INK 9,15:INK 10,16:I
NK 11,17:INK 12,18:INK 13,6:BORDER 15:FOR
i=0 TO 25:OUT &BC00,6:OUT &BD00,i:CALL &BD
19:CALL &BD19:NEXT:RETURN
1260 '                                       [117]
1270 'CLS                                    [467]
1280 '                                       [117]
1290 FOR i=398 TO 32 STEP -2:MOVE 0,i:DRAW  [4758]
 638,i,0:NEXT:FOR i=16 TO 0 STEP 2:MOVE 0,
i:DRAW 638,i,0:NEXT:RETURN
1300 '                                       [117]
1310 'FENSTER                                [1072]
1320 '                                       [117]
1330 GOSUB 1020                              [863]
1340 WINDOW#1,1,20,1,9:PAPER#1,13:CLS#1     [2135]
1350 PAPER 13:PEN 2:LOCATE 3,5:PRINT "POWE  [4716]
R :":LOCATE 3,7:PRINT "TIME        sec."
1360 PEN 11:FOR i=1 TO 20:LOCATE i,1:PRINT  [8301]
 CHR$(143):LOCATE i,9:PRINT CHR$(143):NEXT
:FOR i=2 TO 8:LOCATE 1,i:PRINT CHR$(143):L
OCATE 20,i:PRINT CHR$(143):NEXT
1370 MOVE 0,398:DRAW 638,398,5:DRAW 638,25  [3299]
4,5:DRAW 0,254,4:DRAW 0,398,4
1380 MOVE 32,384:DRAW 606,384,4:DRAW 606,2  [4554]
70,4:DRAW 32,270,5:DRAW 32,384,5
1390 ¦SET,2:¦WINDOW,40,30,2,12:¦FBLOCK,pow  [3358]
er:p=40
1400 MOVE 320,314:DRAW 576,314,4            [1306]
1410 FOR i=320 TO 576 STEP 32:MOVE i,314:D  [2891]
RAWR 0,-4,4:NEXT
1420 PEN 3:PAPER 0:LOCATE 1,25:PRINT">";:L  [4360]
OCATE 20,25:PRINT"<";
```

Listing Joymatte

```
1430 RETURN                                  [555]
1440 '                                       [117]
1450 'Time                                   [362]
1460 '                                       [117]
1470 erg(a)=p:IF a THEN a=a-1:PAPER 13:PEN  [7306]
 4:LOCATE 11,7:PRINT STRING$(4-LEN(STR$(a)
),"0");MID$(STR$(a),2):RETURN ELSE e=1:RET
URN
1480 '                                       [117]
1490 'Jogging                                [153]
1500 '                                       [117]
1510 sp(0)=&942F+2000:sp(1)=&93EF+2000:sp(  [3449]
2)=&93AF+2000
1520 adr(0)=27000:adr(1)=28280:adr(2)=2700  [2111]
0:adr(3)=29560
1530 xb=4:yb=16:gfl=1:x=0:y=0:GOSUB 1050:a  [4726]
$="    Jogging":GOSUB 3280
1540 LOCATE 2,24:FOR i=1 TO 18:PEN RND*12+  [3333]
1:PRINT CHR$(143);:NEXT
1550 ¦SET,1:¦WINDOW,4,184,72,8              [2272]
1560 ¦SET,0:¦WINDOW,32,103,16,80            [2169]
1570 GOSUB 1030                              [869]
1580 '                                       [117]
1590 IF JOY(0)=8 AND fl<>1 THEN fl=1:GOSUB  [2177]
 1660
1600 IF e THEN GOTO 1810                     [515]
1610 IF JOY(0)=4 AND fl<>0 THEN fl=0:GOSUB  [2293]
 1660
1620 t=t+1:IF t>=(73-p)\(5+add) THEN t=0:I  [2178]
F p>40 THEN p=p-1
1630 ¦SET,2:¦POS,p,30:¦FBLOCK,power         [2673]
1640 GOTO 1590                               [317]
1650 '                                       [117]
1660 ¦SET,0:¦FBLOCK,adr(n):n=n+1:IF n=4 TH  [3332]
EN n=0
1670 SOUND 1,0,4,,,,30:IF p<72 THEN p=p+1   [2487]
1680 ¦SET,2:¦POS,p,30:¦FBLOCK,power         [2673]
1690 ¦SET,1:¦LROTATE:CALL &BD19:¦LROTATE:C  [4476]
ALL &BD19:¦LROTATE
1700 RETURN                                  [555]
1710 '                                       [117]
1720 'START                                  [115]
1730 '                                       [117]
1740 PEN 4:PAPER 0:LOCATE 1,12:PRINT"Ready  [8861]
..":¦SET,3:¦WINDOW,0,88,28,8:¦YZOOM,4:FOR
i=1 TO 1000:NEXT:FOR i=1 TO 32:SOUND 1,i*5
,3:¦SCROLLUP:CALL &BD19:CALL &BD19:NEXT
1750 LOCATE 1,12:PRINT"Set...":¦SET,3:¦WIN  [9374]
DOW,0,88,28,8:¦YZOOM,4:FOR i=1 TO 1000:NEX
T:FOR i=1 TO 32:SOUND 1,i*10,3:¦SCROLLUP:C
ALL &BD19:CALL &BD19:NEXT
1760 LOCATE 1,12:PRINT"Go!!!":¦SET,3:¦WIND  [10315]
OW,0,88,20,8:¦YZOOM,4:FOR i=1 TO 1000:NEXT
:FOR i=1 TO 32:SOUND 1,i*20,3:¦SCROLLUP:CA
LL &BD19:CALL &BD19:NEXT:SOUND 1,0,8,0,1,,
2:SOUND 1,0,15,0,1,,30
1770 RETURN                                  [555]
1780 '                                       [117]
1790 'Score                                  [513]
1800 '                                       [117]
1810 NIX=REMAIN(0):wert!=0:FOR i=0 TO VAL(  [5307]
t$):wert!=wert!+erg(i)-40:NEXT:wert!=wert!
/VAL(t$):wert!=wert!*100
1820 LOCATE 2,12:PAPER 0:PEN 4:PRINT"End !  [6945]
!":¦SET,3:¦WINDOW,4,88,24,8:¦YZOOM,4:FOR i
=1 TO 700:NEXT
1830 SC=0:FOR i=1 TO 32:¦SCROLLDOWN:CALL &  [4243]
BD19:CALL &BD19:SOUND 1,0,5,I MOD 15,1,,10
:NEXT
1840 LOCATE 2,12:PRINT"Score:":¦SET,3:¦WIN  [7323]
DOW,4,88,24,8:¦YZOOM,4:LOCATE 2,16:sc=sc+I
NT(wert!):IF sc<0 THEN sc=sc+2^16
1850 sc$=STRING$(7-LEN(STR$(sc)),"0")+MID$  [5377]
(STR$(sc),2):PRINT sc$:¦WINDOW,4,120,24,8:
¦YZOOM,4
1860 MOVE 24,226:DRAW 224,226,7:DRAW 224,1  [6082]
00,7:DRAW 24,100,4:DRAW 24,226,4:GOSUB 311
0:RETURN
1870 '                                       [117]
1880 'Rumpfbeugen                            [1115]
1890 '                                       [117]
1900 sp(0)=&934F+2000:sp(1)=&92EF+2000:sp(  [2565]
2)=&928F+2000
1910 adr(0)=27000:adr(1)=28920:adr(2)=3084  [3800]
0:adr(3)=28920
1920 GOSUB 1020:xb=6:yb=16:gfl=0:GOSUB 105  [5234]
0:a$="   Rumpfbeugen":GOSUB 3280
```

Listing Joymatte

```
1930 ¦SET,0:¦WINDOW,32,103,24,80:GOSUB 103 [2803]
0
1940 IF JOY(0)=3 AND fl<>1 THEN fl=1:GOSUB [3376]
 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
1950 IF e THEN GOTO 1810                   [515]
1960 IF JOY(0)=19 AND fl<>0 THEN fl=0:GOSU [6824]
B 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
1970 t=t+1:IF t>=(73-p)\(3+add) THEN t=0:I [1407]
F p>40 THEN p=p-1
1980 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
1990 GOTO 1940                             [343]
2000 ¦SET,0:¦FBLOCK,adr(n):n=n+1:IF n=4 TH [3332]
EN n=0
2010 SOUND 1,0,4,,,,30:IF p<72 THEN p=p+1  [2487]
2020 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2030 FOR i=1 TO 100:NEXT                   [936]
2040 RETURN                                [555]
2050 '                                     [117]
2060 'Liegestuetz                          [2126]
2070 '                                     [117]
2080 sp(0)=&921F+2000:sp(1)=&91AF+2000:sp( [2542]
2)=&913F+2000
2090 adr(0)=27000:adr(1)=29240:adr(2)=3178 [2330]
0:adr(3)=adr(1)
2100 GOSUB 1020:xb=7:yb=16:gfl=0:GOSUB 105 [5583]
0:a$="  Liegestuetz":GOSUB 3280
2110 ¦SET,0:¦WINDOW,32,103,28,80:GOSUB 103 [2418]
0
2120 IF JOY(0)=3 AND fl<>1 THEN fl=1:GOSUB [3376]
 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
2130 IF e THEN GOTO 1810                   [515]
2140 IF JOY(0)=19 AND fl<>0 THEN fl=0:GOSU [6824]
B 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
2150 t=t+1:IF t>=(73-p)\(3+add) THEN t=0:I [1407]
F p>40 THEN p=p-1
2160 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2170 GOTO 2120                             [307]
2180 '                                     [117]
2190 'Jump                                 [512]
2200 '                                     [117]
2210 sp(0)=&942F+2000:sp(1)=&8FF7+2000:sp( [2416]
2)=&90FF+2000
2220 adr(0)=27000:adr(1)=29304:adr(2)=3160 [3270]
8:adr(3)=29304:adr(4)=adr(0)
2230 GOSUB 1020:x=4:y=8:gfl=2:xb=4:yb=16:G [5255]
OSUB 1050:a$="     Springen":GOSUB 3280
2240 ¦SET,5:¦WINDOW,32,184,40,8:¦GETSPR,33 [2485]
912
2250 ¦SET,0:¦WINDOW,32,95,24,96            [2426]
2260 jump=1:GOSUB 1030:jump=0:fl=0         [1359]
2270 IF JOY(0)=16 AND fl=0 THEN fl=1:GOSUB [1993]
 2330
2280 IF e THEN GOTO 1810                   [515]
2290 IF JOY(0)=3 AND fl=1 THEN fl=0:GOSUB  [2309]
2430
2300 t=t+1:IF t>=(73-p)\(5+add) THEN t=0:I [2178]
F p>40 THEN p=p-1
2310 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2320 GOTO 2270                             [365]
2330 jx=-8:sw=-8:¦SET,0:FOR i=1 TO 4       [2031]
2340 ¦POS,32+i*4,95+jx:¦FBLOCK,adr(i)      [2261]
2350 jx=jx+sw:IF jx=-16 THEN sw=8          [1495]
2360 IF p<72 THEN p=p+1                    [993]
2370 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2380 ¦SET,5:¦FBLOCK,33912:¦SET,0           [1252]
2390 FOR II=1 TO 50:NEXT                   [875]
2400 SOUND 1,0,3,,,,30                     [1217]
2410 NEXT                                  [350]
2420 RETURN                                [555]
2430 jx=-8:sw=-8:¦SET,0:FOR i=3 TO 0 STEP  [2035]
-1
2440 ¦POS,32+i*4,95+jx:¦FBLOCK,adr(i)      [2261]
2450 jx=jx+sw:IF jx=-16 THEN sw=8          [1495]
2460 IF p<72 THEN p=p+1                    [993]
2470 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2480 ¦SET,5:¦FBLOCK,33912:¦SET,0           [1252]
2490 FOR II=1 TO 50:NEXT                   [875]
2500 SOUND 1,0,3,,,,30                     [1217]
2510 NEXT                                  [350]
2520 RETURN                                [555]
2530 '                                     [117]
2540 'Sit-up                               [105]
2550 '                                     [117]
2560 sp(0)=&8F01+2000:sp(1)=&8EBB+2000:sp( [2330]
2)=&8E75+2000
2570 adr(0)=27000:adr(1)=28400:adr(2)=2980 [2216]
0:adr(3)=adr(1)
```

Listing Joymatte

```
2580 GOSUB 1020:x=0:y=0:xb=5:yb=14:gfl=0:G [6073]
OSUB 1050:a$="      Sit-up":GOSUB 3280
2590 ¦SET,0:¦WINDOW,32,114,20,70           [2283]
2600 GOSUB 1030:fl=0                       [995]
2610 IF JOY(0)=19 AND fl<>1 THEN fl=1:GOSU [4441]
B 2000:FOR i=1 TO 200:NEXT:GOSUB 2000
2620 IF e THEN GOTO 1810                   [515]
2630 IF JOY(0)=3 AND fl<>0 THEN fl=0:GOSUB [3455]
 2000:FOR i=1 TO 200:NEXT:GOSUB 2000
2640 t=t+1:IF t>=(73-p)\(4+add) THEN t=0:I [2082]
F p>40 THEN p=p-1
2650 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2660 GOTO 2610                             [345]
2670 '                                     [117]
2680 'Kniebeugen                           [458]
2690 '                                     [117]
2700 sp(0)=&942F+2000:sp(1)=&8F47+2000:sp( [2717]
2)=&90BF+2000
2710 adr(0)=27000:adr(1)=28280:adr(2)=2956 [2909]
0:adr(3)=adr(0)
2720 GOSUB 1020:xb=4:yb=16:GOSUB 1050:a$=" [5778]
   Kniebeugen":GOSUB 3280
2730 ¦SET,0:¦WINDOW,32,103,16,80           [2169]
2740 GOSUB 1030                            [869]
2750 IF JOY(0)=12 AND fl<>1 THEN fl=1:GOSU [4201]
B 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
2760 IF e THEN GOTO 1810                   [515]
2770 IF JOY(0)=28 AND fl<>0 THEN fl=0:GOSU [5107]
B 2000:FOR i=1 TO 100:NEXT:GOSUB 2000
2780 t=t+1:IF t>=(73-p)\(5+add) THEN t=0:I [2178]
F p>40 THEN p=p-1
2790 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2800 GOTO 2750                             [389]
2810 '                                     [117]
2820 'Stuetz-huepfen                       [475]
2830 '                                     [117]
2840 sp(0)=&921F+2000:sp(1)=&8F87+2000:sp( [2525]
2)=&904F+2000
2850 adr(0)=27000:adr(1)=29240:adr(2)=3178 [2330]
0:adr(3)=adr(1)
2860 GOSUB 1020:xb=7:yb=16:GOSUB 1050:a$=" [5961]
 Stuetz-huepfen":GOSUB 3280
2870 ¦SET,0:¦WINDOW,32,103,28,80           [2010]
2880 GOSUB 1030                            [869]
2890 IF JOY(0)=15 AND fl=1 THEN fl=0:GOSUB [8155]
 2000:FOR i=1 TO 100:NEXT:GOSUB 2000:FOR i
=1 TO 50:NEXT
2900 IF e THEN GOTO 1810                   [515]
2910 IF JOY(0)=3 AND fl=0 THEN fl=1:GOSUB  [4118]
2000:FOR i=1 TO 100:NEXT:GOSUB 2000:FOR i=
1 TO 50:NEXT
2920 t=t+1:IF t>=(73-p)\(2+add) THEN t=0:I [2354]
F p>40 THEN p=p-1
2930 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
2940 GOTO 2890                             [321]
2950 '                                     [117]
2960 'Entspannen                           [1326]
2970 '                                     [117]
2980 sp(0)=&8E35+2000:sp(1)=&8DF5+2000:sp( [2028]
2)=&8DB5+2000
2990 adr(0)=27000:adr(1)=28280:adr(2)=adr( [2936]
0):adr(3)=29560
3000 GOSUB 1020:xb=4:yb=16:gfl=1:GOSUB 105 [5559]
0:a$="   Entspannen":GOSUB 3280
3010 ¦SET,0:¦WINDOW,32,103,16,80:GOSUB 103 [2651]
0
3020 IF JOY(0)<>0 AND fl=0 THEN fl=1:GOSUB [1713]
 2000
3030 IF e THEN GOTO 1810                   [515]
3040 IF JOY(0)=0 AND fl=1 THEN fl=0:GOSUB  [1732]
2000
3050 t=t+1:IF t>=(73-p)\(4+add) THEN t=0:I [2082]
F p>40 THEN p=p-1
3060 ¦SET,2:¦POS,p,30:¦FBLOCK,power        [2673]
3070 GOTO 3020                             [461]
3080 '                                     [117]
3090 'Musik1                               [1101]
3100 '                                     [117]
3110 ENV 1,1,14,1,1,0,10,14,-1,4:ENV 2,1,1 [2477]
4,1,14,-1,4
3120 DATA 17,284,25,10,716,25,17,268,25,10 [5980]
,716,25,17,284,25,10,716,25,17,319,25,10,7
16,25,2,716,25,2,716,25,2,716,25,2,716,25
3130 DATA 17,358,25,10,536,25,17,358,25,10 [5558]
,536,25,17,426,25,10,536,25,17,358,50,10,4
78,25,2,478,25,17,319,25,10,478,25,2,478,2
5,2,478,25
```

Listing Joymatte

```
3140 DATA 17,284,25,10,716,25,17,268,25,10 [5980]
,716,25,17,284,25,10,716,25,17,319,25,10,7
16,25,2,716,25,2,716,25,2,716,25,2,716,25
3150 DATA 17,358,25,10,536,25,17,358,25,10 [5558]
,536,25,17,426,25,10,536,25,17,358,50,10,4
78,25,2,478,25,17,319,25,10,478,25,2,478,2
5,2,478,25
3160 DATA 17,358,23,10,426,25,17,358,23,10 [8714]
,426,25,17,358,23,10,426,25,17,358,48,10,4
26,25,2,426,25,17,358,25,10,426,25,2,478,2
5,17,426,25,10,426,25
3170 DATA 17,358,25,10,536,25,17,426,25,10 [6380]
,536,25,17,358,25,10,536,25,17,284,48,10,5
68,25,2,568,25,17,284,25,10,568,25,2,568,2
5,2,568,25
3180 DATA 17,268,23,10,638,25,17,268,23,10 [5973]
,638,25,17,268,23,10,638,25,17,239,48,10,5
68,25,2,568,25,17,239,25,10,568,25,2,568,2
5,17,239,25,10,568,25
3190 DATA 17,213,23,10,638,25,17,213,23,10 [6084]
,638,25,17,213,23,10,638,25,17,213,50,10,4
78,25,2,478,25,17,239,50,10,478,25,2,478,2
5,2,478,25
3200 DATA 99,99,99,99,99,99                [915]
3210 RESTORE 3120                          [743]
3220 PEN 2:LOCATE 9,12:PRINT">Press Fire<" [5558]
:MOVE 252,204:DRAW 638,204,4:DRAWR 0,22,5:
DRAWR -386,0,5:DRAWR 0,-22,4
3230 READ s,h,l:IF s=99 THEN GOTO 3210 ELS [2949]
E IF JOY(0)=16 THEN CALL &BCA7:RETURN
3240 IF s=17 THEN SOUND s,h\2,1,14:READ s, [4659]
h,l:SOUND s,h\2,L,0,1:SOUND 4,H,L,14
3250 IF s=1 THEN SOUND s,h\2,1,14          [1464]
3260 IF s=2 THEN SOUND s,h\2,1,0,1:SOUND 4 [2677]
,H,L,0,2
3270 GOTO 3230                             [443]
3280 WINDOW 1,20,1,15:CLS:WINDOW 1,20,1,25 [8249]
:GOSUB 1340:PAPER 13:POKE &A611,13:CALL &A
610:PEN 4:LOCATE 3,3:GOSUB 4000:POKE &A611
,0:CALL &A610:RETURN
3290 '                                     [117]
3300 'Music2                               [1116]
3310 '                                     [117]
3320 RESTORE 3350                          [737]
3330 ENV 1,1,13,26,13,-1,5                 [1038]
3340 ENV 2,1,13,26,13,-1,30:ENV 3,5,3,4,15 [2165]
,-1,10
3350 DATA 17,568,30,10,1703,150            [982]
3360 DATA 1,338,30,1,284,30,1,284,30,1,284 [13247]
,30,17,284,60,10,1276,120,1,319,30,1,319,3
0,17,338,30,10,1136,120,1,379,30,1,379,30,
1,426,30,17,426,120,10,1703,120,17,568,30,
10,1703,150,1,338,30,1,284,30,1,284,30,1,2
84,30,17,253,60,10,1276,120,1,284,30,1,319
3370 DATA 30,17,338,30,10,1136,120,1,379,3 [11550]
0,1,379,30,1,338,30,17,338,120,10,1703,120
,17,568,30,10,1703,150,1,338,30,1,284,30,1
,284,30,1,284,30,17,213,60,10,1276,120,1,2
25,30,1,253,30,17,284,30,10,1136,120,1,319
,30,1,319,30,1,338,30,17,338,120,10,1703,1
20
3380 DATA 17,568,30,10,1703,150,1,338,30,1 [8966]
,284,30,1,284,30,1,284,30,17,253,60,10,127
6,120,1,284,30,1,319,30,17,338,30,10,1136,
120,1,379,30,1,379,30,1,426,30,17,426,120,
10,1012,120,17,451,90,10,1351,90
3390 DATA 17,506,90,10,1276,120,1,568,30,1 [9736]
7,426,30,10,1136,120,1,379,30,1,426,30,1,4
51,30,17,426,120,10,1703,120,17,426,30,10,
1703,60,1,426,30,17,379,60,10,1136,240,1,5
68,30,1,568,15,1,568,45,1,568,60,1,568,30
3400 DATA 17,319,60,10,1276,120,1,338,30,1 [11267]
,379,30,17,426,60,10,1703,120,1,284,30,1,2
84,30,17,284,30,10,1351,120,1,284,60,1,284
,30,17,284,60,10,1276,120,1,319,30,1,338,3
0,17,379,120,10,1136,240,1,0,60,1,284,30,1
,284,30
3410 DATA 17,284,60,10,1136,240,1,190,30,1 [10334]
,190,15,1,190,15,1,190,30,1,190,30,1,0,30,
1,190,30,17,213,60,10,1136,240,1,225,30,1,
253,30,1,284,120,17,319,30,10,1517,60,1,33
8,30,17,379,60,10,1136,120,1,426,30,1,451,
30,17,426,120,10,1703,120
3420 DATA 17,319,30,10,1517,60,1,338,30,17 [5853]
,379,120,10,1136,180,1,426,30,1,451,30,17,
426,180,10,1703,180,99,99,99,99,99,99
3430 READ s,h,l:IF s=99 THEN RESTORE 3350: [4154]
ENV 2,13,1,8,13,-1,8:GOTO 3430
```

Listing Joymatte

```
3440 IF s=1 THEN SOUND 1,h\2,1,0,1         [1792]
3450 IF s=17 THEN SOUND 17,h\2,1,0,1:READ  [5270]
s2,h2,12:SOUND s2,h2\2,12,14:SOUND 4,H2\4,
12,0,2
3460 ON SQ(1) GOSUB 3430:RETURN            [1300]
3470 RESTORE 3350:GOTO 3430                [1013]
3480 '                                     [117]
3490 'Titel                                [643]
3500 '                                     [117]
3510 TAGOFF:GOSUB 1020:PEN 1:PAPER 0:MODE  [2332]
0
3520 PEN 1:LOCATE 2,3:a$="CPC":GOSUB 4000: [4410]
|SET,3:|WINDOW,4,16,13,10:|XZOOM,6:|YZOOM,
4
3530 PEN 2:LOCATE 4,10:a$="Fitness":GOSUB  [4678]
4000:|SET,3:|WINDOW,12,72,29,10:|XZOOM,2:|
YZOOM,4
3540 PEN 3:LOCATE 2,17:a$="Center":GOSUB 4 [6605]
000:|SET,3:|WINDOW,4,128,25,10:|XZOOM,3:|Y
ZOOM,4
3550 PEN 11:FOR i=1 TO 20:LOCATE i,1:PRINT [8138]
 CHR$(143):LOCATE i,25:PRINT CHR$(143);:NE
XT:FOR i=2 TO 24:LOCATE 1,i:PRINT CHR$(133
):LOCATE 20,i:PRINT CHR$(138):NEXT
3560 MOVE 0,0:DRAW 0,398,4:DRAW 638,398,4: [1774]
DRAW 638,0,5:DRAW 0,0,5
3570 MOVE 16,16:DRAW 16,382,5:DRAW 622,382 [3995]
,5:DRAW 622,16,4:DRAW 16,16,4
3580 LOCATE 3,23:PEN 5:PRINT"by Klaus  Ebe [3797]
rle"
3590 DIM name$(13)                         [1014]
3600 ORIGIN 0,0,64,640,34,32:i=24:|SET,3:| [3186]
WINDOW,8,164,69,26:num=0
3610 name$(0)="    in 1988":name$(1)="Spec [12984]
ial thanks to":name$(2)="Carsten&Wolfgang"
:name$(3)="Special greetings":name$(4)=" t
o Khalid,Joerg,":name$(5)=" Kamil and Volk
er":name$(6)="    MC-routines":name$(7)="
   from"
3620 name$(8)="    BLUE-BASIC":name$(9)="  [10939]
   Music:":name$(10)=" Zwischen boese":nam
e$(11)="      und gut":name$(12)=" Urlaub im
 Sueden":name$(13)="     Flashdance"
3630 GOSUB 1250:GOSUB 3320:PLOT 700,700,5  [1889]
3640 ON SQ(1) GOSUB 3430                   [866]
3650 MOVE 64,I:TAG:PRINT name$(num);:TAGOF [5747]
F:i=i+1:|SCROLLUP:IF i=56 THEN i=24:num=nu
m+1:IF num=14 THEN num=1
3660 IF INKEY$="" THEN 3650 ELSE ORIGIN 0, [2700]
0,0,640,400,0
3670 MOVE 32,30:DRAW 600,30,0:LOCATE 3,21: [8295]
PEN 0:PRINT STRING$(17,CHR$(143));:LOCATE
3,23:PEN 0:PRINT STRING$(17,CHR$(143));:LO
CATE 3,22:PRINT STRING$(17,CHR$(143));
3680 LOCATE 4,23:PEN 4:A$="Player(1-9) :": [2805]
GOSUB 4000
3690 LOCATE 17,23:PRINT"   ";:LOCATE 17,23 [3199]
:MAXi=2:GOSUB 3900
3700 spieler=VAL(a$):IF spieler>9 OR (spie [4995]
ler=0 AND UPPER$(a$)<>"O") THEN GOTO 3690
ELSE IF A$="O" THEN 3770
3710 FOR NUM=0 TO spieler-1:MOVE 32,30:DRA [9972]
W 600,30,0:LOCATE 2,23:PEN 0:PRINT STRING$
(17,CHR$(143));:LOCATE 2,22:PRINT STRING$(
17,CHR$(143));
3720 LOCATE 2,23:PEN 4:A$="Name"+STR$(NUM+ [5398]
1)+" :":GOSUB 4000:LOCATE 11,23:MAXi=7:GOS
UB 3900
3730 name$(NUM)=UPPER$(a$):punkte(NUM)=0   [1448]
3740 NEXT                                  [350]
3750 FOR NUM=spieler TO 9:name$(NUM)="":NE [2561]
XT
3760 GOTO 3800                             [481]
3770 MOVE 32,30:DRAW 600,30,0:LOCATE 3,23: [9999]
PEN 0:PRINT STRING$(16,CHR$(143));:LOCATE
3,22:PRINT STRING$(16,CHR$(143));:LOCATE 2
,23:PEN 4:A$="Filename:":GOSUB 4000:LOCATE
 11,23:MAXi=8:GOSUB 3900
3780 CALL &BCA7:i=0:|OPEN,@i,@a$:IF i<>0 T [8678]
HEN MODE 2:INK 1,0:PEN 1:PRINT:PRINT " Thi
s is your directory:":PRINT:PRINT:WINDOW 2
,79,2,24:CAT:CALL &BB06:WHILE INKEY$<>"":W
END:RUN
3790 OPENIN "!"+a$:INPUT#9,spieler:FOR i=0 [4321]
 TO 8:INPUT#9,alt(i):INPUT#9,name$(i):NEXT
:CLOSEIN:RESTORE 3350
```

Listing Joymatte

```
3800 ON SQ(1) GOSUB 3430:MOVE 32,30:DRAW 6 [13554]
00,30,0:LOCATE 2,23:PEN 0:PRINT STRING$(16
,CHR$(143));:PEN 4:LOCATE 6,21:A$="<E>asy.
....":GOSUB 4000:LOCATE 6,22:a$="<N>ormal.
..":GOSUB 4000:LOCATE 6,23:A$="<D>ifficult
":GOSUB 4000:
3810 in$="0":WHILE INSTR("END",in$)=0:in$= [1984]
UPPER$(INKEY$):IF in$="" THEN in$="0"
3820 WEND                                   [390]
3830 IF in$="E" THEN t$="010":add=-1       [1558]
3840 IF in$="N" THEN t$="020":add=0        [2298]
3850 IF in$="D" THEN t$="040":add=1        [664]
3860 CALL &BCA7:RETURN                     [958]
3870 '                                     [117]
3880 'Input                                [213]
3890 '                                     [117]
3900 LOCATE POS(#0),VPOS(#0):PRINT "<";:po [3349]
salt=POS(#0):VPOSALT=VPOS(#0)
3910 WHILE INKEY$<>"":WEND                 [1786]
3920 i=0:a$="":in$="":WHILE i<MAXi AND IN$ [3037]
<>CHR$(13)
3930 in$=UPPER$(INKEY$):IF in$<>"" THEN IF [7984]
 ASC(in$)<>127 AND in$<>CHR$(13) AND ASC(i
n$)>31 THEN PRINT CHR$(8);in$;"<";:i=i+1:a
$=a$+in$:GOTO 3950
3940 IF i>0 AND in$<>CHR$(13) AND in$=CHR$ [5727]
(127) THEN PRINT CHR$(8);" ";CHR$(8);CHR$(
8);"<";:a$=MID$(a$,1,LEN(a$)-1):i=i-1
3950 WEND                                  [390]
3960 FOR i1=i+posalt-1 TO Posalt STEP -1:L [7393]
OCATE i1,VPOSALT:PRINT "<":CALL &BD19:LOCA
TE i1,VPOSALT:PRINT " ":NEXT:LOCATE posalt
-1,vposalt:PRINT"<":RETURN
3970 '                                     [117]
3980 '3D                                   [315]
3990 '                                     [117]
4000 PLOT 700,700,7:xn=POS(#0):yn=VPOS(#0) [12543]
:MOVE (xn-1)*32+4,398-((yn-1)*16)-2:TAG:PR
INT a$;:LOCATE xn,yn:TAGOFF:IF yn<>25 THEN
 PRINT CHR$(22);CHR$(1);a$;CHR$(22);CHR$(0
):RETURN ELSE PRINT CHR$(22);CHR$(1);a$;CH
R$(22);CHR$(0);:RETURN
4010 '                                     [117]
4020 'ENDE                                 [715]
4030 '                                     [117]
4040 MODE 0:GOSUB 1250:FOR I=0 TO 13:INK I [13977]
,0:NEXT:BORDER 0:INK 15,1,26:INK 14,0:LOCA
TE 5,3:PEN 15:PAPER 14:PRINT"Please  wait"
:MOVE 320,0:RAD:FOR i!=-PI/2 TO PI/2+0.1 S
TEP 0.1:MOVE 320+320*SIN(i!),320*COS(i!):D
RAW 320,0,4:NEXT
4050 WINDOW 5,8,23,25:PAPER 13:CLS:WINDOW  [7410]
9,12,17,25:PAPER 2:CLS:WINDOW 13,16,20,25:
PAPER 3:CLS:PAPER 0
4060 WINDOW 1,20,1,25:PEN 5:PAPER 13:LOCAT [6951]
E 6,24:PRINT "3":PAPER 2:LOCATE 10,19:PRIN
T"1":PAPER 3:LOCATE 14,22:PRINT"2":PAPER 0
4070 |SET,0:adr(0)=27000:entspl=&8E35+2000 [3036]
4080 |WINDOW,0,0,4,16:|FBLOCK,entspl:|XZOO [4860]
M,4:|YZOOM,5:|GETSPR,adr(0)
4090 |WINDOW,16,96,16,80:|FBLOCK,adr(0)    [2165]
4100 |WINDOW,32,48,16,80:|FBLOCK,adr(0)    [1906]
4110 |WINDOW,48,72,16,80:|FBLOCK,adr(0)    [2419]
4120 WINDOW 1,4,1,10:PAPER 0:CLS:WINDOW 1, [3127]
20,1,25
4130 LOCATE 1,1:PEN 2:A$="*** Hall of Fame [6796]
 ***":GOSUB 4000:A$="*":FOR I=2 TO 25:LOCA
TE 1,I:GOSUB 4000:LOCATE 20,I:GOSUB 4000:N
EXT
4140 LOCATE 2,2:GOSUB 4000:LOCATE 19,2:GOS [4809]
UB 4000:LOCATE 2,24:GOSUB 4000:LOCATE 19,2
4:GOSUB 4000
4150 FOR I=2 TO 3:LOCATE I,25:GOSUB 4000:L [3748]
OCATE I+16,25:GOSUB 4000:NEXT
4160 FOR I=0 TO SPIELER-2                  [1077]
4170 FOR J=I TO SPIELER-1                  [1121]
4180 IF PUNKTE(I)<PUNKTE(J) THEN HILF=PUNK [9811]
TE(I):HILF$=NAME$(I):PUNKTE(I)=PUNKTE(J):P
UNKTE(J)=HILF:NAME$(I)=NAME$(J):NAME$(J)=H
ILF$
4190 NEXT                                  [350]
4200 NEXT                                  [350]
4210 PRINT CHR$(22);CHR$(1);:PEN 5:LOCATE  [11573]
5,12:PRINT NAME$(2):LOCATE 9,6:PRINT NAME$
(0):LOCATE 13,9:PRINT NAME$(1):PRINT CHR$(
22);CHR$(0);:LOCATE 5,3:PAPER 0:PRINT STRI
NG$(12," "):GOSUB 1250
```

Listing Joymatte

```
4220 fl=0:RESTORE 4410:ENV 1,1,14,2,14,-1, [3629]
7:ENT 1,1,1,1,0,30:ENT 2,1,-1,1,1,0,30
4230 NIX=1:C=2:ON SQ(4) GOSUB 4490         [1566]
4240 FOR i=1 TO 500:NEXT:WHILE INKEY$<>"": [2329]
WEND:WHILE INKEY$="":WEND
4250 GOSUB 1020                            [863]
4260 LOCATE 13,9:PRINT STRING$(7," ");:WIN [5410]
DOW 5,16,2,25:PAPER 0:CLS:WINDOW 1,20,1,25
4270 FOR i!=-PI/2 TO PI/2+0.1 STEP 0.1:MOV [7339]
E 320+320*SIN(i!),320*COS(i!):DRAW 320,0,4
:NEXT:PRINT CHR$(22);CHR$(1);
4280 GOSUB 1250:FOR i=0 TO 8:IF name$(I)<> [10348]
"" THEN LOCATE 4,3+i*2:PEN i+1:PRINT name$
(i)+STRING$(9-LEN(name$(i))," ");STRING$(6
-LEN(STR$(punkte(i))),"0");MID$(STR$(punkt
e(i)),2)
4290 IF NAME$(I)="" THEN LOCATE 4,3+i*2:PE [9351]
N i+1:PRINT STRING$(7,".")+STRING$(2," ");
STRING$(6-LEN(STR$(punkte(i))),"0");MID$(S
TR$(punkte(i)),2)
4300 NEXT                                  [350]
4310 LOCATE 5,25:PEN 7:PRINT">Press  Key<" [4195]
;CHR$(22);CHR$(0);
4320 LOCATE 2,24:PAPER 2:PRINT STRING$(18, [8748]
" "):PEN 3:LOCATE 3,24:PRINT"Insert Disk/C
ass":WHILE INKEY$<>"":WEND:WHILE INKEY$=""
:WEND
4330 LOCATE 2,24:PEN 3:PRINT"Filename:     [9868]
    ":LOCATE 11,24:maxi=8:GOSUB 3900:IF I
NSTR(a$,".")<>0 OR INSTR(a$,",")<>0 THEN L
OCATE 2,24:PRINT STRING$(18," "):GOTO 4330
4340 IF a$="" THEN LOCATE 11,24:PRINT"No F [4801]
ile":FOR i=1 TO 1000:NEXT:GOTO 4360
4350 CALL &BCA7:OPENOUT "!"+a$:PRINT #9,sp [5285]
ieler:FOR i=0 TO 8:PRINT#9,punkte(i):PRINT
#9,name$(i):NEXT:CLOSEOUT
4360 PAPER 1:PEN 1:RETURN                  [1558]
4370 '                                     [117]
4380 'SPIELER                              [91]
4390 '                                     [117]
4400 |SET,1:LOCATE 11,7:PEN 4:PRINT t$:a=V [12239]
AL(t$):PAPER 0:LOCATE 2,25:PRINT STRING$(1
8," ");:LOCATE 2,25:A$="Player: "+STRING$(
9-LEN(NAME$(NUM))," ")+NAME$(NUM):GOSUB 40
00:RETURN
4410 DATA 49,213,180,42,536,240,28,268,30, [5434]
4,268,30,4,268,30,4,268,30,4,268,30,4,268,
30,33,239,30,12,268,30,33,268,30,12,268,30
4420 DATA 49,239,176,42,716,240,28,358,30, [6435]
4,358,30,4,358,30,4,358,30,4,358,30,4,358,
30,33,239,30,12,358,30,33,213,30,12,358,30
4430 DATA 49,201,180,42,956,240,28,478,30, [4559]
4,478,30,4,478,30,4,478,30,4,478,30,4,478,
30,33,213,30,12,478,30,33,239,30,12,478,30
4440 DATA 49,213,30,42,638,240,28,319,30,3 [7030]
3,239,30,12,319,30,33,268,120,12,319,30,4,
319,30,4,319,30,4,319,30,33,239,30,12,319,
30,33,268,30,12,319,30
4450 DATA 49,159,180,42,804,240,28,402,30, [6900]
4,402,30,4,402,30,4,402,30,4,402,30,4,402,
30,33,179,30,12,402,30,33,201,30,12,402,30
4460 DATA 49,179,180,42,536,240,28,268,30, [5136]
4,268,30,4,268,30,4,268,30,4,268,30,4,268,
30,33,201,30,12,268,30,33,213,30,12,268,30
4470 DATA 49,239,210,42,602,240,28,301,30, [7725]
4,301,30,4,301,30,4,301,30,4,301,30,4,301,
30,4,301,30,33,268,30,12,301,30,49,239,120
,42,716,120,28,358,30,4,358,30,4,358,30,4,
358,30
4480 DATA 99,99,99,99                      [669]
4490 READ s,h,l:IF s=99 THEN RESTORE 4410: [5891]
IF NIX=0 THEN C=1:GOTO 4550 ELSE NIX=0:GOT
O 4550
4500 IF s=49 THEN SOUND s,h,l,14,,1:READ s [10303]
,h,l:SOUND s,h,l,14,,2:READ s,h,l:IF NIX=1
 THEN SOUND s,h,l ELSE IF fl=0 THEN fl=1:S
OUND s,h,l,0,1 ELSE IF fl=1 THEN fl=0:SOUN
D s,h\C,l,0,1
4510 IF s=1 THEN SOUND 1,h,l,14,,1         [1444]
4520 IF s=2 THEN SOUND 2,H,L,14,,2         [2517]
4530 IF s=4 THEN IF NIX=1 THEN SOUND s,h,l [4956]
 ELSE IF fl=0 THEN fl=1:SOUND s,h,l,0,1 EL
SE IF fl=1 THEN fl=0:SOUND s,h\C,l,0,1
4540 IF s=33 THEN SOUND s,h,l,14,,1:READ s [9096]
,h,l:IF NIX=1 THEN SOUND s,h,l ELSE IF fl=
0 THEN fl=1:SOUND s,h,l,0,1 ELSE IF fl=1 T
HEN fl=0:SOUND s,h\C,l,0,1
4550 ON SQ(4) GOSUB 4490:RETURN            [1472]
```

Listing Joymatte

```
10 ' FITSP.BAS                                        [402]
20 '                                                  [117]
1000 DATA 00,00,C0,00,00,40,C0,80,00,84,C0 [2421]
,48,00,C0,C0,C0,00,84,C0,48,00,40
1010 DATA 0C,80,00,00,C0,00,10,CC,CC,20,30 [3543]
,CC,CC,20,64,CC,CC,20,20,CC,CC,20
1020 DATA A0,54,28,A0,00,FC,28,00,00,A8,28 [3999]
,00,54,00,28,00,03,00,03,00,00,C0
1030 DATA 00,00,40,C0,80,00,84,C0,48,00,C0 [3741]
,C0,C0,00,84,C0,48,00,40,0C,80,00
1040 DATA 00,C0,00,00,10,CC,CC,20,10,CC,CC [4077]
,30,10,CC,CC,98,10,CC,CC,10,50,54
1050 DATA 28,50,00,54,3C,00,00,54,14,00,00 [2820]
,54,00,28,00,03,00,03,00,40,80,00
1060 DATA 00,C0,C0,00,CC,40,48,84,80,40,C0,C0 [2958]
,80,40,48,84,80,00,84,48,00,00,40
1070 DATA 80,00,10,CC,CC,20,10,CC,CC,20,10 [3717]
,CC,CC,20,10,CC,CC,20,50,54,28,A0
1080 DATA 00,54,28,00,00,54,28,00,00,54,28 [2009]
,00,00,03,03,00,00,00,00,00,00,00
1090 DATA 00,00,00,00,00,00,00,00,00,00,00 [1902]
,00,00,00,00,00,00,C0,80,00,00,40
1100 DATA C0,C0,00,CC,C8,C0,C0,44,CC,C8,48 [3335]
,48,CC,98,C8,48,48,6C,98,60,48,80
1110 DATA 6C,CC,30,40,00,3C,6C,98,A0,01,3C [4850]
,3C,3C,3C,29,14,3C,3C,3C,29,00,00
1120 DATA 00,00,00,00,00,00,80,00,00,00,40 [4162]
,C0,00,00,00,C0,84,80,00,00,C0,C0
1130 DATA 80,00,00,C0,48,C0,00,CC,C0,84,00 [2485]
,00,98,C8,C0,00,44,98,64,00,00,CC
1140 DATA CC,30,70,00,CC,CC,98,20,00,9C,3C [3373]
,28,00,01,14,3C,3C,3C,29,00,3C,3C
1150 DATA 3C,29,40,C0,00,00,00,C0,84,80,00 [3774]
,00,C0,C0,C0,00,00,C0,84,08,00,00
1160 DATA 40,C0,80,00,00,00,CC,00,00,00,44 [2715]
,98,88,00,00,CC,98,CC,00,00,CC,98
1170 DATA CC,00,00,CC,98,CC,00,00,44,98,88 [2691]
,00,00,14,78,3C,00,01,14,3C,3C,3C
1180 DATA 29,00,3C,3C,3C,29,00,00,00,00,00 [3674]
,C0,80,00,40,C0,48,00,40,C0,C0,80
1190 DATA 40,C0,0C,00,00,C0,C0,00,00,40,80 [2929]
,00,00,CC,CC,00,44,CC,30,70,44,CC
1200 DATA 30,20,44,CC,CC,88,00,CC,CC,00,00 [3499]
,14,28,00,00,00,3C,00,00,14,28,00
1210 DATA 00,01,03,00,00,00,00,00,00,00,00 [2675]
,00,00,00,00,00,00,00,00,00,00,00
1220 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1230 DATA 00,00,00,00,00,00,40,C0,00,00,00 [3208]
,00,00,C0,C0,80,00,00,44,CC,C0,C0
1240 DATA 80,00,14,CC,CC,C0,C0,80,00,14,6C [3443]
,64,84,84,80,00,14,3C,64,84,84,80
1250 DATA 00,00,3C,64,84,C0,00,00,16,6C,20 [2626]
,00,80,00,00,02,00,20,00,80,00,00
1260 DATA 02,00,F0,00,00,00,00,00,00,00,00 [4608]
,C0,80,00,40,C0,48,00,40,C0,C0,80
1270 DATA 40,C0,0C,00,00,C0,C0,00,00,40,80 [3606]
,00,00,CC,CC,00,44,CC,64,88,44,98
1280 DATA 64,88,44,30,CC,88,00,64,CC,00,50 [3094]
,14,28,00,00,00,3C,00,00,14,28,00
1290 DATA 00,01,03,00,C3,F3,0F,F3,CF,F3,33 [2968]
,F3,C3,F3,0F,F3,CF,F3,33,F3,C3,F3
1300 DATA 0F,F3,CF,F3,33,F3,00,00,00,00,00 [3563]
,00,00,00,00,00,00,00,00,00,00,00
1310 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1320 DATA 00,00,00,00,00,00,00,00,00,00,00 [2729]
,00,00,00,00,40,C0,00,00,00,44,CC
1330 DATA C0,C0,80,00,14,CC,CC,C0,C0,80,00 [2872]
,14,6C,64,C0,C0,80,00,00,3C,64,84
1340 DATA 84,80,00,00,14,64,84,84,80,00,00 [2862]
,16,20,84,C0,00,00,00,02,20,00,80
1350 DATA 00,00,00,02,F0,00,80,00,00,00,00 [3995]
,00,00,00,00,00,00,C0,80,00,40,C0
1360 DATA 48,00,40,C0,C0,80,40,C0,0C,00,00 [3486]
,C0,C0,00,00,40,80,00,00,CC,CC,00
1370 DATA 44,CC,30,70,44,CC,30,20,44,CC,CC [2665]
,88,00,CC,CC,00,00,00,3C,28,00,14
1380 DATA 28,00,00,01,03,00,00,C0,80,00,40 [1810]
,C0,48,00,40,C0,C0,80,40,C0,0C,00
1390 DATA 00,C0,C0,00,00,40,80,00,00,CC,CC [4115]
,00,44,CC,64,88,44,CC,30,88,44,CC
```

Listing Joymatte

```
1400 DATA 98,20,00,CC,CC,70,00,14,28,00,00 [2609]
,00,3C,01,00,00,14,29,00,00,00,29
1410 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1420 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1430 DATA 00,00,00,00,00,00,00,00,00,00,00 [2381]
,00,00,40,C0,00,00,00,00,00,C0,C0
1440 DATA 80,00,00,00,00,C0,C0,80,00,00,44 [4142]
,CC,C0,C0,80,00,14,CC,CC,84,84,80
1450 DATA 14,3C,CC,64,84,84,80,16,3C,98,64 [4219]
,84,C0,00,02,00,30,CC,00,80,00,02
1460 DATA 00,50,A0,00,80,00,00,00,00,00,00 [3365]
,00,00,00,00,00,00,00,00,00,00,00
1470 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1480 DATA 00,00,00,00,00,00,00,00,40,C0,00 [3459]
,00,00,00,00,C0,C0,80,00,00,00,00
1490 DATA C0,C0,80,00,00,44,CC,C0,C0,80,00 [4317]
,14,CC,CC,84,84,80,00,3C,CC,64,84
1500 DATA 84,80,14,3C,98,CC,84,C0,00,16,00 [3733]
,98,CC,00,80,00,02,00,10,88,00,80
1510 DATA 00,02,00,50,A0,00,00,00,00,00,00 [3933]
,00,00,00,00,00,00,00,00,00,00,00
1520 DATA 00,00,00,00,00,00,00,00,00,00,00 [2211]
,00,00,00,00,00,00,00,00,00,00,00
1530 DATA 00,00,00,40,C0,00,00,00,00,00,C0 [3761]
,C0,80,00,00,00,00,C0,C0,80,00,00
1540 DATA 44,CC,C0,C0,80,00,14,CC,CC,84,84 [3352]
,80,00,3C,CC,64,84,84,80,00,3C,CC
1550 DATA 64,84,C0,00,14,28,CC,64,00,80,00 [4190]
,16,00,44,20,00,80,00,02,00,00,20
1560 DATA 00,00,00,02,00,00,F0,00,00,00,00 [3393]
,00,00,00,00,00,00,00,00,00,00,00
1570 DATA 00,00,00,00,00,00,00,00,00,00,00 [1669]
,00,00,00,00,00,C0,80,00,00,00,40
1580 DATA C0,C0,00,00,CC,C8,C0,C0,00,44,CC [3242]
,C8,48,48,00,CC,CC,C8,48,48,00,CC
1590 DATA CC,60,48,80,00,44,CC,20,40,00,00 [3457]
,14,6C,20,40,00,00,14,28,20,00,00
1600 DATA 00,14,28,20,00,00,00,14,28,A0,00 [3454]
,00,00,01,03,00,00,00,00,00,00,00
1610 DATA 00,00,00,00,00,80,00,00,00,00,40 [3234]
,C0,00,00,00,00,C0,84,80,00,00,00
1620 DATA C0,C0,80,00,00,00,C0,48,C0,00,00 [2620]
,CC,C0,84,00,00,00,98,C8,C0,00,00
1630 DATA 44,98,64,00,00,00,44,CC,30,00,00 [2241]
,00,44,CC,98,20,00,00,00,9C,28,70
1640 DATA 00,00,00,14,28,00,00,00,00,14,28 [3270]
,00,00,00,00,14,28,00,00,00,00,01
1650 DATA 03,00,00,00,00,C0,80,00,00,00,40 [3468]
,C0,48,00,00,40,C0,C0,80,00,00
1660 DATA 40,C0,0C,00,00,00,00,C0,C0,00,00 [2147]
,00,00,44,88,00,00,00,00,CC,64,00
1670 DATA 00,00,44,CC,64,88,00,00,44,CC,64 [2574]
,88,00,00,44,CC,64,88,00,00,00,CC
1680 DATA 64,00,00,00,00,14,A0,00,00,00,00 [2444]
,14,28,00,00,00,00,14,28,00,00,00
1690 DATA 00,14,28,00,00,00,00,01,03,00,00 [3362]
,00,00,40,C0,00,00,C0,84,80,00,C0
1700 DATA C0,C0,00,C0,84,08,00,40,C0,80,00 [3142]
,44,88,00,00,CC,64,00,44,CC,30,88
1710 DATA 44,CC,98,20,44,CC,CC,20,B0,CC,CC [3457]
,50,00,3C,7C,00,14,3C,FC,00,3C,00
1720 DATA 54,A8,28,00,00,A8,03,00,00,03,40 [3364]
,C0,00,00,C0,84,80,00,C0,C0,C0,00
1730 DATA C0,84,08,00,40,C0,80,00,00,44,88 [3739]
,00,00,CC,64,00,44,98,64,88,44,30
1740 DATA CC,88,10,64,CC,88,A0,CC,CC,70,00 [3505]
,3C,28,00,54,BC,28,00,FC,00,3C,00
1750 DATA A8,00,14,28,03,00,00,03,00,C0,80 [2692]
,00,40,C0,48,00,40,C0,C0,80,40,C0
1760 DATA 0C,00,00,C0,C0,00,00,44,88,00,00 [3112]
,CC,64,00,44,CC,64,88,44,CC,64,88
1770 DATA 44,CC,64,88,00,CC,64,00,00,14,A0 [2771]
,00,00,14,28,00,00,14,28,00,00,14
1780 DATA 28,00,00,01,03,00                       [505]
1790 RESTORE 1000:FOR a!= 36277 TO 37998: [3596]
READ a$:POKE a!,VAL("&"+a$):NEXT
1795 SAVE"fitsp.bin",b,36277,1722          [1819]
```

Listing Joymatte

```
1000 ' fitmc.bas                                      [741]
1010 '                                                [117]
1020 DATA 21,4A,9C,01,4E,9C,CD,D1,BC,C9,00 [3282]
,00,00,00,77,9C,= 1576
1030 DATA C3,C1,9C,C3,45,9F,C3,E9,9F,C3,38 [2650]
,9D,C3,1A,9D,C3,= 2535
1040 DATA FC,9D,C3,BB,9D,C3,80,9E,C3,56,9E [2101]
,C3,7B,9D,C3,4E,= 2616
1050 DATA 9D,C3,02,9D,C3,21,9D,4F,50,45,CE [2125]
,59,5A,4F,4F,CD,= 1872
1060 DATA 57,49,4E,44,4F,D7,53,45,D4,49,4E [2557]
,49,D4,53,43,52,= 1632
1070 DATA 4F,4C,4C,55,D0,53,43,52,4F,4C,4C [3457]
,44,4F,57,CE,58,= 1515
1080 DATA 5A,4F,4F,CD,47,45,54,41,44,D2,4C [2130]
,52,4F,54,41,54,= 1490
1090 DATA C5,50,4F,D3,47,45,54,53,50,D2,46 [3126]
,42,4C,4F,43,CB,= 1725
1100 DATA 00,FE,02,C2,F8,9C,21,5A,BB,46,C5 [2900]
,36,C9,DD,5E,00,= 2001
1110 DATA DD,56,01,1A,13,EB,5E,23,56,EB,11 [1882]
,00,C0,47,CD,77,= 1642
1120 DATA BC,F5,CD,7A,BC,F1,C1,21,5A,BB,78 [1879]
,77,D2,F8,9C,DD,= 2766
1130 DATA 6E,02,DD,66,03,AF,77,C9,DD,6E,02 [2073]
,DD,66,03,3E,FF,= 1909
1140 DATA 77,C9,B7,CA,79,A0,FD,2A,DE,A1,FD [2533]
,6E,00,FD,66,0E,= 2396
1150 DATA DD,5E,00,DD,56,01,C3,38,9E,C9,21 [1791]
,60,A1,22,DE,A1,= 1940
1160 DATA C9,B7,CA,79,A0,FD,2A,DE,A1,FD,6E [2411]
,00,FD,66,0E,DD,= 2498
1170 DATA 5E,00,DD,56,01,C3,4F,A0,B7,CA,79 [4050]
,A0,DD,6E,00,FE,= 2087
1180 DATA 0E,D2,80,A0,26,00,11,60,A1,19,22 [2184]
,DE,A1,C9,FE,02,= 1723
1190 DATA C2,79,A0,FD,2A,DE,A1,DD,7E,00,FE [2833]
,C8,D2,72,A0,6F,= 2549
1200 DATA FD,77,D6,DD,7E,02,FE,50,D2,72,A0 [2845]
,67,FD,77,C8,CD,= 2633
1210 DATA CA,9F,FD,75,00,FD,74,0E,C9,00,00 [2415]
,3E,7E,32,A0,9D,= 1870
1220 DATA FD,2A,DE,A1,FD,7E,E4,32,A8,9D,3D [2103]
,FE,50,D2,72,A0,= 2539
1230 DATA FD,66,C8,FD,6E,D6,CD,CA,9F,FD,46 [3426]
,F2,C5,22,79,9D,= 2772
1240 DATA 7E,FD,77,BA,54,5D,23,01,01,00,0D [2928]
,ED,B0,FD,7E,BA,= 1889
1250 DATA 12,2A,79,9D,CD,2D,9F,C1,10,E2,C9 [3046]
,FD,2A,DE,A1,FD,= 2314
1260 DATA 7E,D6,FD,86,F2,3D,6F,FD,7E,C8,FD [2946]
,86,E4,3D,67,CD,= 2704
1270 DATA CA,9F,54,5D,CD,2D,9F,EB,FD,46,F2 [2533]
,0E,00,FD,7E,E4,= 2368
1280 DATA 32,E6,9D,C5,E5,01,01,00,ED,B8,D1 [2714]
,62,6B,CD,37,9F,= 2119
1290 DATA C1,10,F0,FD,46,E4,AF,12,1B,10,FC [2384]
,C9,FD,2A,DE,A1,= 2367
1300 DATA FD,66,C8,FD,6E,D6,CD,CA,9F,54,5D [3749]
,CD,2D,9F,FD,46,= 2607
1310 DATA F2,0E,00,FD,7E,E4,32,1C,9E,C5,E5 [2260]
,01,50,00,ED,B0,= 2019
1320 DATA D1,62,6B,CD,2D,9F,C1,10,F0,FD,46 [3146]
,E4,AF,12,13,10,= 2051
1330 DATA FC,C9,FD,5E,62,FD,56,70,FD,7E,F2 [2472]
,E5,FD,4E,E4,06,= 2764
1340 DATA 00,ED,B0,E1,01,00,08,09,38,05,3D [1871]
,C2,3B,9E,C9,01,= 1391
1350 DATA 50,C0,09,C3,4A,9E,FE,02,C2,79,A0 [3173]
,DD,6E,00,DD,66,= 2093
1360 DATA 01,4E,06,00,23,5E,23,56,D5,1B,EB [2739]
,09,CB,FE,E1,CD,= 1706
1370 DATA D4,BC,D2,80,A0,EB,DD,6E,02,DD,66 [2403]
,03,73,23,72,C9,= 2257
1380 DATA FD,2A,DE,A1,FD,7E,E4,FD,66,C8,84 [3177]
,67,25,FD,6E,D6,= 2689
1390 DATA CD,CA,9F,22,C4,9F,FD,6E,E4,7D,DD [3277]
,46,00,05,C8,05,= 2172
1400 DATA 28,03,85,10,FD,F5,85,32,C8,9F,F1 [2998]
,16,00,5F,2A,C4,= 1828
1410 DATA 9F,19,22,C6,9F,ED,5B,C4,9F,FD,7E [2313]
,F2,F5,FD,46,E4,= 2675
1420 DATA DD,4E,00,1A,E6,55,77,0D,20,06,DD [2718]
,4E,00,C3,E7,9E,= 1693
```

Listing Joymatte

```
1430 DATA CB,27,B6,77,2B,0D,28,03,C3,C3,9E [3697]
,DD,4E,00,1A,E6,= 1745
1440 DATA AA,CB,3F,77,0D,28,13,1A,E6,AA,B6 [2613]
,77,2B,0D,28,03,= 1453
1450 DATA C3,DE,9E,05,28,11,1B,C3,C0,9E,05 [1768]
,28,0A,1B,DD,4E,= 1590
1460 DATA 00,1A,E6,55,C3,C3,9E,F1,3D,CA,26 [3257]
,9F,F5,2A,C6,9F,= 2234
1470 DATA CD,2D,9F,EB,2A,C4,9F,CD,2D,9F,EB [2350]
,ED,53,C4,9F,22,= 2394
1480 DATA C6,9F,F1,C3,BC,9E,3A,C8,9F,FD,77 [3039]
,E4,C9,01,00,08,= 2366
1490 DATA 09,D0,01,50,C0,09,C9,7C,D6,08,67 [2040]
,FE,C0,D0,B7,01,= 1987
1500 DATA 50,C0,ED,42,C9,FE,01,C2,79,A0,FD [2784]
,2A,DE,A1,FD,7E,= 2563
1510 DATA D6,FD,5E,F2,83,3D,32,C8,9F,6F,FD [2731]
,66,C8,CD,CA,9F,= 2636
1520 DATA 22,C4,9F,DD,46,00,FD,7E,F2,05,C8 [2383]
,4F,05,28,03,81,= 1762
1530 DATA 10,FD,F5,81,ED,4B,C8,9F,32,C8,9F [3488]
,F1,81,6F,FD,66,= 2559
1540 DATA C8,CD,CA,9F,22,C6,9F,FD,46,F2,0E [3091]
,00,C5,DD,7E,00,= 2280
1550 DATA 2A,C4,9F,ED,5B,C6,9F,F5,FD,4E,E4 [2323]
,AF,47,7A,F6,C0,= 2692
1560 DATA 57,ED,B0,2A,C6,9F,CD,37,9F,F1,22 [2420]
,C6,9F,3D,C2,90,= 2349
1570 DATA 9F,2A,C4,9F,CD,37,9F,22,C4,9F,C1 [3770]
,10,CF,3A,C8,9F,= 2197
1580 DATA FD,77,F2,C9,00,00,00,00,00,00,7D [2108]
,00,00,5F,54,E6,= 1349
1590 DATA F8,6F,AF,67,44,4D,29,29,09,29,7B [3062]
,E6,07,87,87,87,= 1679
1600 DATA 84,67,4A,09,7C,C6,C0,67,C9,FE,04 [2889]
,C2,79,A0,FD,2A,= 2164
1610 DATA DE,A1,DD,7E,00,FE,C9,D2,72,A0,FD [2597]
,77,F2,DD,7E,02,= 2632
1620 DATA FE,51,D2,72,A0,FD,77,E4,DD,7E,04 [2716]
,FE,C8,D2,72,A0,= 2708
1630 DATA FD,77,D6,DD,7E,06,FE,50,D2,72,A0 [4112]
,FD,77,C8,FD,6E,= 2692
1640 DATA D6,FD,66,C8,F5,CD,CA,9F,F1,FD,75 [2852]
,00,FD,74,0E,FD,= 2827
1650 DATA 4E,E4,81,FE,51,D2,72,A0,FD,4E,F2 [4150]
,FD,7E,D6,81,FE,= 2803
1660 DATA C9,D2,72,A0,26,00,6C,54,DD,5E,00 [2891]
,DD,46,02,C9,FD,= 1977
1670 DATA 7E,E4,32,5A,A0,FD,7E,F2,E5,01,04 [2837]
,00,EB,ED,B0,EB,= 2392
1680 DATA E1,01,00,08,09,38,04,3D,20,EE,C9 [2156]
,01,50,C0,09,C3,= 1312
1690 DATA 67,A0,21,8E,A0,CD,E5,A0,C9,21,A5 [2401]
,A0,CD,E5,A0,C9,= 2546
1700 DATA 21,B8,A0,CD,E5,A0,C9,21,CD,A0,CD [4244]
,E5,A0,C9,56,61,= 2548
1710 DATA 6C,75,65,20,6F,75,74,20,6F,66,20 [2611]
,73,63,72,65,65,= 1509
1720 DATA 6E,2E,2E,2E,00,50,61,72,61,6D,65 [2780]
,74,65,72,20,65,= 1310
1730 DATA 72,72,6F,72,2E,2E,2E,00,56,61,6C [2734]
,75,65,20,72,61,= 1343
1740 DATA 6E,67,65,20,65,72,72,6F,72,2E,2E [2816]
,2E,00,53,70,72,= 1347
1750 DATA 69,74,65,20,64,6F,65,73,6E,27,74 [1780]
,20,65,78,69,73,= 1519
1760 DATA 74,2E,2E,2E,00,7E,FE,00,CD,5A,BB [3493]
,C8,23,C3,E5,A0,= 1935
2900 '                                        [117]
2910 MEMORY &9C3F                             [472]
2920 ZEILE = 1000 : SCHRITT = 10              [1491]
2930 ADR = &9C40 : LAST = &A0F0               [1946]
2940 FOR I=1 TO 16                            [317]
2950      READ BYTE$                          [603]
2960      POKE ADR,VAL("&"+BYTE$)             [745]
2970      SUM=SUM+PEEK(ADR)                   [1499]
2980      ADR=ADR+1                           [392]
2990 NEXT                                     [350]
3000 READ CHECKSUM$:CHECKSUM=VAL(MID$(CHEC [3979]
KSUM$,3))
3010 IF SUM<>CHECKSUM THEN PRINT "FEHLER I [2721]
N ZEILE";ZEILE
3020 IF ADR<LAST THEN SUM=0:ZEILE=ZEILE+SC [2270]
HRITT:GOTO 2940
3030 SAVE"fitmc.bin",b,40000,&4B1             [1968]
```

Listing Joymatte

CPC-Listingservice

Eingabekontrolle für Listings

für 464-664-6128

Die Fehlersuche von abgetippten Listings ist manchmal eine lästige und nervenaufreibende Sache. Wenn Sie auch dieser Meinung sind, dann können wir Ihnen eine Hilfe anbieten.

Auf dieser Seite befinden sich drei Programme, die für jeden CPC-Rechner abgestimmt sind. Der "Checksummer" hilft Ihnen bei jeder Eingabe von Programmlistings.
Bevor Sie sich das Programm, das für Ihren Rechner geeignet ist, vornehmen, sollten noch einige "Dinge" beachtet werden:

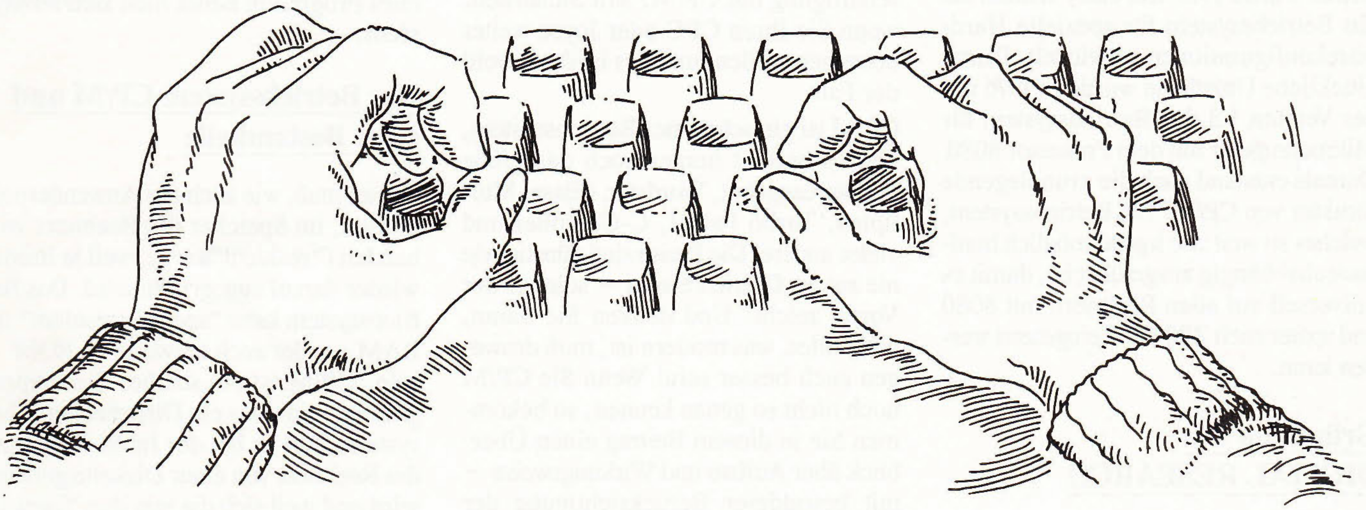
1. Das abgetippte Programm vor dem ersten Start abspeichern, da der Checksummer Maschinensprache enthält.

2. Um den Checksummer benutzen zu können, genügt es, das Programm einmal durch RUN zu starten. Es wird selbständig der Maschinencode eingelesen und aufgerufen.

3. Jedesmal, wenn die ENTER oder RETURN-Taste gedrückt wird, erscheint eine Quersumme der eingegebenen Zeile. Z.B. wenn ohne Zeile die ENTER-Taste gedrückt wird, erhalten Sie in eckigen Klammern am Ende der Zeile eine <0>. Dadurch lassen sich eingetippte Listings gleich auf Fehler überprüfen.

4. Weitere Möglichkeiten: Möchten Sie ein Listing mit dem Checksummer ausdrucken, so läßt sich auch das bewerkstelligen. Z.B. mit Icheck,x,y. Wobei x = Ausgabekanal und y = Zeilennummer ist. Vor dem Druck sollte man mit WIDTH x die maximale Breite (x) einer Zeile festlegen. Mit Hilfe der RSX-Befehle ON und OFF kann der Checksummer ein- bzw. ausgeschaltet werden.

(Thomas Fippl/cd)



Version 464

```
10 REM *** CHECKSUM 464 ***                [1071]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,98,aa,c3           [799]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 06,dd,11,3a,bd,01,03,00           [923]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a2           [1665]
220 DATA c1,cd,a3,e7,e5,4e,23,46           [1294]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,3c,c4,e5,09,e3,cd,63           [1318]
250 DATA e1,21,a4,ac,cd,7a,a5,e1           [1223]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,96,f2,e3,cd,f6           [973]
280 DATA a5,cd,4e,c3,e1,7e,a7,c8           [738]
290 DATA cd,98,a5,cd,4e,c3,18,f5           [1601]
300 DATA 3a,24,ac,d6,08,47,7e,a7           [1761]
310 DATA c8,cd,45,e1,23,10,f7,c9           [1394]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,8a,ff,4f           [1826]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,5c,c3,3e,5b,cd,5c,c3           [1002]
430 DATA cd,79,ee,3e,5d,c3,5c,c3           [1170]
```

Version 664

```
10 REM *** CHECKSUM 664 ***                [1262]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,02,ac,c3           [333]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 4c,cb,11,5b,bd,01,03,00           [1095]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a9           [1682]
220 DATA c1,cd,69,e8,e5,4e,23,46           [1382]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,75,c4,e5,09,e3,cd,59           [798]
250 DATA e2,21,8a,ac,cd,7a,a5,e1           [1228]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,58,f3,e3,cd,f6           [1104]
280 DATA a5,cd,9b,c3,e1,7e,a7,c8           [1523]
290 DATA cd,98,a5,cd,9b,c3,18,f5           [1069]
300 DATA 3a,09,ac,d6,08,47,7e,a7           [1172]
310 DATA c8,cd,22,e2,23,10,f7,c9           [1848]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,ab,ff,4f           [1559]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,a3,c3,3e,5b,cd,a3,c3           [1480]
430 DATA cd,49,ef,3e,5d,c3,a3,c3           [1389]
```

Version 6128

```
10 REM *** CHECKSUM 6128 ***               [945]
20 MEMORY &A4FF                            [463]
30 FOR a%=&A500 TO &A607                   [903]
40 READ byte$                              [603]
50 POKE a%,VAL("&"+byte$)                  [938]
60 NEXT                                    [350]
70 PRINT                                   [361]
80 PRINT"CHECKSUM v2"                      [1533]
90 CALL &A500:|ON:NEW                      [978]
100 :                                      [174]
110 DATA 21,09,a5,01,0d,a5,c3,d1           [1000]
120 DATA bc,00,00,00,00,18,a5,c3           [757]
130 DATA 2a,a5,c3,2f,a5,c3,43,a5           [1681]
140 DATA 4f,ce,4f,46,c6,43,48,45           [1162]
150 DATA 43,cb,00,00,cf,02,ac,c3           [333]
160 DATA a8,a5,21,27,a5,18,03,21           [1100]
170 DATA 24,a5,28,06,cd,00,b9,c3           [1425]
180 DATA 49,cb,11,5e,bd,01,03,00           [1424]
190 DATA ed,b0,c9,4f,cd,00,b9,0d           [1382]
200 DATA 28,08,0d,20,ea,dd,7e,02           [481]
210 DATA 18,04,7b,11,01,00,cd,a6           [1661]
220 DATA c1,cd,64,e8,e5,4e,23,46           [812]
230 DATA 23,5e,23,56,e1,78,b1,c8           [1226]
240 DATA cd,72,c4,e5,09,e3,cd,54           [1100]
250 DATA e2,21,8a,ac,cd,7a,a5,e1           [1228]
260 DATA 18,e2,e5,cd,ba,a5,e3,cd           [1468]
270 DATA 98,a5,cd,53,f3,e3,cd,f6           [1107]
280 DATA a5,cd,98,c3,e1,7e,a7,c8           [1357]
290 DATA cd,98,a5,cd,98,c3,18,f5           [1651]
300 DATA 3a,09,ac,d6,08,47,7e,a7           [1172]
310 DATA c8,cd,1d,e2,23,10,f7,c9           [1782]
320 DATA cd,24,a5,f5,c5,d5,e5,cd           [1019]
330 DATA ba,a5,cd,f6,a5,e1,d1,c1           [1941]
340 DATA f1,c9,eb,1b,af,47,67,6f           [1774]
350 DATA 2f,32,23,a5,13,1a,d6,30           [1279]
360 DATA 38,04,fe,0a,38,f6,1a,13           [1209]
370 DATA a7,c8,4f,3a,23,a5,a1,fe           [1573]
380 DATA 20,28,f3,79,fe,22,20,07           [1570]
390 DATA 3a,23,a5,2f,32,23,a5,3a           [1615]
400 DATA 23,a5,a7,79,c4,ab,ff,4f           [1559]
410 DATA ad,07,6f,09,18,d8,3e,20           [791]
420 DATA cd,a0,c3,3e,5b,cd,a0,c3           [1906]
430 DATA cd,44,ef,3e,5d,c3,a0,c3           [1650]
```

CP/M Intern I

Eine Einführung in das Betriebssystem CP/M

Der CPC ist "von Hause aus" ein BASIC-Rechner. Dieses dient sowohl als Programmiersprache wie auch als Betriebssystem. Der JOYCE mit dem Programm LOCOSCRIPT gilt als Textverarbeitungsrechner.

Oft wird dabei übersehen, daß beide vollwertige CP/M-Rechner sind. Beim CPC läuft die Version CP/M 2.2 auf allen Modellen mit mindestens einem Diskettenlaufwerk. Auf dem CPC 6128 läuft außerdem noch die Version 3 (CP/M Plus), die auch Verwendung beim Joyce findet.

CP/M (Control Program for Microcomputers wurde 1973 von Gary Kildall als **ein** Betriebssystem für spezielle Hardwarekonfigurationen entwickelt. Durch glückliche Umstände wurde es 1976 mit der Version 1.3 **das** Betriebssystem für Mikrocomputer mit dem Prozessor 8080. Damals entstand auch die grundlegende Struktur von CP/M. Ein Betriebssystem, welches so weit wie irgend möglich hardwareunabhängig ausgeführt ist, damit es universell auf allen Rechnern mit 8080 und später auch Z80 CPU eingesetzt werden kann.

Gründung von DIGITAL RESEARCH

1976 wurde auch die Vertriebsfirma DIGITAL RESEARCH gegründet. Es entstanden dann weitere Versionen (bis 2.2), die insgesamt auf Computertypen von fast 100 verschiedenen Herstellern in einer Stückzahl von weit über 200000 liefen. Weiter entstand ein riesiges Angebot von Programmen, welche problemlos (gegebenenfalls nach einer kleinen Anpassung) auf allen diesen Computertypen lauffähig waren. Der Glanz von Digital Research schien unübertreffbar – das Elend aber folgte auf dem Fuße. 1982 erschien eine ganz neue Version CP/M 3, auch CP/M Plus genannt. Das erwartete weltweite Echo aber blieb aus, denn ein neuer Stern am Computerhimmel war aufgegangen: Der IBM PC mit dem 16-Bit-Prozessor 8088 (Vorläufer 8086). Auch er brauchte (für seine damals bescheidenen 64 kByte RAM) ein Betriebssystem. Das **hätte** eine CP/M-Version von Digital Research werden können. Durch eine Verkettung merkwürdiger Umstände wurde aber nichts daraus. MICROSOFT mit MS-DOS (PC-DOS) machte das Rennen. Daran konnten auch die nachträglichen Anstrengungen von Digital Research (CP/M 86) nichts mehr ändern.

Heute gibt es außer den CPCs und dem Joyce keine CP/M-Rechner mehr, und dadurch ist auch CP/M selbst nicht mehr aktuell.

Lohnt sich dann überhaupt noch die Beschäftigung mit CP/M? Mit Sicherheit, wenn Sie Ihren CPC oder Joyce weiter betreiben wollen, und das ist doch wohl der Fall.

CP/M ist ein sehr gutes Betriebssystem, und es besteht immer noch das große Softwareangebot, Wordstar, dBase, Multiplan, Turbo Pascal, C-Compiler und vieles andere. Die Preise sind günstig wie nie zuvor. Greifen Sie zu – solange der Vorrat reicht. Und denken Sie daran, nicht alles, was modern ist, muß deswegen auch besser sein! Wenn Sie CP/M noch nicht so genau kennen, so bekommen Sie in diesem Beitrag einen Überblick über Aufbau und Wirkungsweise – mit besonderer Berücksichtigung der Verhältnisse beim CPC. Aber auch wenn Sie schon Bescheid wissen, werden Sie vielleicht das eine oder andere erfahren, was Ihnen noch nicht bekannt war. Wer tiefer einsteigen will, insbesondere etwa Programmierung unter CP/M verstehen und vielleicht auch selbst betreiben möchte, muß weitere Fachliteratur studieren. Eine geeignete Auswahl ist am Ende des Berichts angegeben. Hinweise zur **praktischen** Handhabung von CP/M finden Sie in dem Beitrag "Die residenten und transienten Commands von CP/M".

```
            +---------------------+ FFFF
            !        BIOS         !
            !                     !
            !                     !
            +---------------------+
            !        BDOS         !
            !                     !
FBASE       +---------------------+
            !        CCP          !
CBASE       +---------------------+
            !                     !
            !        TPA          !
            !                     !
TBASE       +---------------------+ 0100
            !      ZEROPAGE       !
BOOT        +---------------------+
```

Abb.1: Allgemeine Speicheraufteilung unter dem Betriebssystem CP/M 2.2

Ein Rechner ohne Betriebssystem ist für den Benutzer praktisch wertlos. Er will zum Beispiel wissen, welche Programme er auf einer Diskette zur Verfügung hat, und will eines davon starten. Wenn dieses Programm läuft, greift es auf die Rechnerperipherie wie Tastatur, Bildschirm, Drucker und Diskettenlaufwerk zu. Wenn diese Arbeiten (und noch andere) mehr rationell erledigt werden sollen, so braucht man ein universelles Überwachungs- und Steuerprogramm, welches einen vernünftigen Rechnerbetrieb überhaupt erst ermöglicht. Ein solches Programm nennt man **Betriebssystem.**

Das Betriebssystem CP/M und seine Bestandteile

Dieses muß, wie auch das Anwenderprogramm, im Speicher des Rechners vorhanden ("resident") sein, weil ja immer wieder darauf zugegriffen wird. Das Betriebssystem kann "speicherresident" im RAM – oder auch teilweise im ROM – sein. CP/M ist ein solches Systemprogramm, und zwar ein Diskettenbetriebssystem, weil es bei der Inbetriebnahme des Rechners von einer Diskette geladen wird und weil sich die von ihm "verwalteten" Dateien auf einer Diskette (nicht Kassette!) befinden müssen.

CP/M besteht aus drei Teilen:

1. Dem CCP (Console Command Processor)
2. Dem BDOS (Basic Disk Operating System)
3. Dem BIOS (Basic Input Output System)

Im Fall 2 und 3 hat das Wort "Basic" nichts mit der Programmiersprache "BASIC" zu tun, sondern ist sinngemäß mit "Basis" zu übersetzen. Der **CCP** ist die "Benutzeroberfläche" von CP/M. Wenn das System gestartet wird – wir werden auf dieses "Booten", wie es genannt wird, noch eingehen – und jedesmal, wenn ein Programm beendet ist, meldet sich CP/M über den CCP durch Angabe des "angemeldeten" Laufwerks mit dem folgenden "Prompt" (Zeichen ">"). Der Cursor dahinter zeigt, daß der CCP bereit ist, "Kommandos" vom Benutzer entgegenzunehmen. Das kann ein "residentes" Kommando sein (zum

Beispiel "DIR"), welches im CCP "eingebaut" (also vorgesehen ist) und dann sofort ausgeführt wird. Es kann auch ein "transientes" Kommando sein. Dieses besteht dann in dem Aufruf eines Programms, welches sich auf der angemeldeten Diskette befindet. Diese Programme sind entweder Hilfsprogramme für das System – zum Beispiel PIP.COM – oder Anwenderprogramme. Ein solches Programm wird "transient", das heißt, "vorübergehend" von der Diskette in den dafür vorgesehenen Teil des RAM geladen ("TPA" = Transient Program Area) und anschließend ausgeführt. CP/M ist mit dem CCP ein Betriebssystem "klassischer Bauart". Es verlangt Eingaben in Textform, hat also keine "grafische Benutzeroberfläche". Das gilt übrigens auch für modernere Betriebssysteme wie MS-DOS oder sogar OS/2. Bei diesen gibt es dann allerdings Ergänzungsprogramme mit grafischen Symbolen und Mausbetätigung.

Wenn ein transientes Programm gestartet ist, hat der CCP seine Schuldigkeit getan. Jetzt tritt der Hauptteil von CP/M, das **BDOS** in Aktion.

BDOS, die Schnittstelle

Über das BDOS laufen alle Ein- und Ausgaben und alle Diskettenzugriffe eines Programms. BDOS ist also "Schnittstelle" (das ist die etwas unglückliche Übersetzung für "interface" zwischen Programm und Betriebssystem). Diese Operationen werden durch standardisierte Befehlsfolgen, die sogenannten "BDOS-Funktionen", ausgeführt. In jedem CP/M-Programm ist dann also die Ausgabe eines Zeichens auf dem Bildschirm oder der Zugriff auf eine Diskettendatei in genau gleicher Weise programmiert.

Das BIOS

Die Hardware der einzelnen Rechnertypen ist aber durchaus unterschiedlich. Die Ansteuerung von Tastatur, Bildschirm und Drucker erfolgt über Ports (oder auch Speicherstellen) mit den verschiedensten Adressen. Der Zugriff auf Diskettenlaufwerke geschieht über einen FDC (Floppy-Disk-Controller), von dem es auch die verschiedensten Typen gibt. Kurz und gut, allein mit einem universellen BDOS ist es nicht getan. Man benötigt noch eine weitere Schnittstelle zwischen BDOS und Hardware – nämlich das **BIOS.** CCP und BDOS sind also für alle CP/M-Computer (mit 8080 beziehungsweise Z80 CPU) einheitlich programmiert, während das BIOS vom Hersteller an die Hardware angepaßt werden muß. Damit die Universalität von CP/M gewahrt bleibt, muß sich auch der Übergang vom BDOS zum BIOS in einer festgelegten Weise abspielen. Während das Anwenderprogramm die erforderlichen BDOS-Funktionen aufruft, ruft seinerseits BDOS durch standardisierte Befehlsfolgen die "BIOS-Routinen" auf. Diese Aufrufe erfolgen über die sogenannte "BIOS-Sprungleiste". Im BIOS finden wir auch eine Reihe von Tabellen, in denen die Kenndaten für das verwendete Diskettenformat festgehalten sind. Dazu gehört zum Beispiel der **DPB** (Disk-Parameter-Block). Wir können uns das Zusammenwirken von CP/M und Anwenderprogramm an einem "Schalenmodell" vorstellen: Die äußerste Schale ist der CCP, der an seiner "Oberfläche" (über Tastatur und Bildschirm) mit dem Benutzer korrespondiert. Die erste innere Schale ist das vom Benutzer aufgerufene Anwenderprogramm.

```
      +-----------------------+ FFFF
      !                       !
      !         BIOS          !
      !                       !
      +-----------------------+ AD00
      !                       !
      !         BDOS          !
      !                       !
FBASE +-----------------------+ 9F00
      !                       !
      !         CCP           !
      !                       !
CBASE +-----------------------+ 9700
      !                       !
      !                       !
      !                       !
      !         TPA           !
      !                       !
      !                       !
      !                       !
TBASE +-----------------------+ 0100
      !        ZEROPAGE       !
BOOT  +-----------------------+
```

Abb.2: CP/M 2.2 im Speicher des CPC

Darunter liegt als zweite innere Schale das BDOS, welches über seine Oberfläche (BDOS-Funktionen) mit dem Anwenderprogramm korrespondiert. Unter dem BDOS liegt als dritte innere Schale das "allgemeine BIOS", welches wiederum über seine Oberfläche (BIOS-Sprungleiste) mit dem BDOS verkehrt. Der innerste Kern schließlich ist das "individuelle" BIOS, der Teil, der mit der Rechner-Hardware verkehrt. Durch diese Grundkonzeption von CP/M wird erreicht, daß nur der innere BIOS-Kern für den einzelnen Rechner angepaßt programmiert werden muß, während alles andere einheitlich ist. Dadurch ist auch gewährleistet, daß ein CP/M-Anwenderprogramm grundsätzlich auf allen CP/M-Rechnern lauffähig ist.

Auf diese Weise hat CP/M Ruhm, Ansehen und seine außerordentliche Verbreitung gefunden. Außerdem ist das Konzept so gut, daß es von allen folgenden Betriebssystemen im Prinzip übernommen wurde. Um eine unkontrollierte Verbreitung zu vermeiden, wurden von Digital Research CCP und BDOS nur unter strengen Lizenzen an die Computerhersteller weitergegeben. Lediglich für die individuelle BIOS-Anpassung wurden Vorschläge und Listings veröffentlicht. Allerdings haben die Hersteller dann ihre BIOS-Versionen meist wieder als strenges Geheimnis gehütet. Wer dagegen heute kommentierte Listings von CCP und BDOS veröffentlicht (für die Version 2.2 ist dies bereits geschehen), muß kaum eine Intervention von Digital Research befürchten.

CP/M im Speicher

CP/M ist speicherresident, wie wir gehört haben, und Abbildung 1 zeigt, wie das bei einem CP/M-Rechner mit der Version 2.2 prinzipiell aussieht. Die CPU kann maximal 64 kByte adressieren, und dann liegt das obere Speicherende bei FFFF (alle Adressenangaben hexadezimal). Das BDOS hat eine Länge von 14 pages, der CCP eine Länge von 8 pages (1 page = 256 Bytes). Das BIOS ist individuell programmiert und hat keine feste Länge. Ein typischer CP/M-Computer hatte zum Beispiel ein BIOS von 4 kByte = 16 pages. Dann wird

```
       +------------------------+-------------------+-------------------+ FFFF
       !        VIDEO-RAM       !    BASIC-ROM      ! AMSDOS-ROM        !
       !                        !                   ! (VDOS)            !
...... +------------------------+-------------------+-------------------+ C000
.      ! Syst.Bereich(Blocksw.) !
.      !                        !
.      +------------------------+ AC00
.      !                        !
.      !                        !
COMMON !                        !
AREA   !         BASIC          !
.      ! Anwender-Bereich       !
.      !                        !
.      !                        !
.      !                        !
...... !                        +-------------------+ 4000
       !                        !                   !
       !                        ! Betriebssyst.     !
  0040 +------------------------+ (unteres) ROM     !
       !       RST-Vektoren     !                   !
       +------------------------+-------------------+ 0000
```

Abb.3a: Der Speicher des CPC unter BASIC

BIOSanfang = F000 FBASE = E200 CBASE = DA00.

Wozu dient die "Zeropage"?

Die BDOS-Funktionen werden vom Anwenderprogramm aufgerufen. Da aber (wegen der unterschiedlichen Länge des BIOS) das BDOS keine festen Adressen im Speicher belegt, kann dieser Aufruf nicht direkt durch einen Sprung in das BDOS erfolgen. Er wird indirekt über eine feste Adresse in der Zeropage (0005) geführt.

Eine weitere wichtige Adresse in der Zeropage ist 0000. Über diese Adresse wird ein Sprung zu der BIOS-Routine "Warmboot" ausgeführt. Der Aufruf erfolgt üblicherweise nach Beendigung eines transienten Programms. Dabei wird der CCP (bei der Version 2.2 auch das BDOS) neu in den Speicher geladen und CCP gestartet.

Warum wird der CCP neu geladen?

Während der Ausführung eines transienten Programms können keine CP/M-Kommandos mehr eingegeben werden. Der Benutzer kann nicht mehr mit dem Betriebssystem, sondern nur noch mit dem laufenden Programm über entsprechende BDOS-Funktionen kommunizieren. Der CCP wird für diesen Zeitraum nicht benötigt, und sein Speicherplatz kann vom transienten Programm genutzt und dabei überschrieben werden. Daher erfolgt sicherheitshalber immer das Nachladen. Mit dem weiteren Inhalt der Zeropage- (Sprungadressen-) Tabellen wollen wir uns hier nicht befassen.

Die TPA von CP/M

Die oben erwähnte **TPA** beginnt immer bei 0100. Hierhin wird jedes unter CP/M laufende Programm durch den CCP geladen und ab Adresse 0100 gestartet. Die TPA kann sich sicher bis CBASE erstrecken. Wenn der CCP während dem Ablauf eines Programms überschrieben wird, kann sich die TPA sogar bis FBASE erstrecken. In unserem Beispiel hat sie damit eine Größe von E200 - 0100 = E100 Bytes, das sind (E=14 14 x 4 = 54 kByte plus 1 page, das heißt 54 1/4 kByte. Beim CPC sieht die Speicherbelegung unter CP/M 2.2 leider viel ungünstiger aus (Abbildung 2). FBASE ist 9F00 und die TPA hat die Größe 9F00 - 0100 = 9E00 Bytes, das sind 9 x 4 = 36 kByte plus 14 pages (3 1/2 kByte), das heißt 39 1/2 kByte. Das ist so wenig, daß kaum ein Anwenderprogramm darin läuft. (Es gibt einige Ausnahmen, zum Beispiel der gute alte Wordstar).

Der CPC ist von Hause aus als grafikfähiger BASIC-Rechner konzipiert. Er besitzt daher ein 16 kByte großes "Video-RAM", da jedes "Pixel" des Bildschirmes mindestens durch ein Bit angesprochen werden muß. Das eigentliche "BASIC" (also das Interpretierprogramm ist in einem ROM von ebenfalls 16 kByte enthalten). Das Betriebssystem, welches ja auch ein BASIC-Rechner benötigt, ist in einem weiteren ROM von 16 kByte enthalten. Berücksichtigt man, daß ROM-Programme immer noch zusätzliche RAM brauchen, so bleibt, wenn man den oben genannten Speicherbedarf addiert, für die BASIC-Anwenderprogramme praktisch nichts übrig.

Was ist die Ursache für diese kleine TPA?

Die AMSTRAD-Entwickler haben sich daher einer Technik bedient, die auch bei anderen Mikrocomputern schon lange verwendet wurde. Es handelt sich um das sogenannte "Bankswitching" oder "Blockswitching". Die CPU kann nur einen Gesamtspeicher von 64 kByte adressieren. Man kann aber einzelne Speicherbereiche (RAM oder ROM) "ausblenden" und dafür andere Bereiche gleicher Größe (ebenfalls RAM oder ROM) "einblenden". Diese Bereiche haben beim CPC eine einheitliche Größe von 16 kByte. Man nennt sie "Blöcke" (daher "Blockswitching"). Die gesamten jeweils adressierten 64 kByte bilden eine "Speicherbank". Nach dem Ausblenden alter und dem Einblenden neuer Blöcke adressiert der Prozessor dann eine neue Speicherbank (daher "Bankswitching"). Diese umschaltbaren Speicherblöcke tauchen neuerdings auch bei den MS-DOS-Rechnern auf, wo sie vielfach "caches" genannt werden. Einige Autoren übersetzen dies mit "Kacheln", was zwar ähnlich klingt, aber leider falsch ist.

Softwaremäßig wird das Blockswitching vom Systemprogramm aus veranlaßt. Die hardwaremäßigen Voraussetzungen, (die zusätzlichen) Speicherblöcke und eine entsprechende Umschaltlogik müssen natürlich vorhanden sein. Das laufende Anwenderprogramm darf von der Umschaltung natürlich nichts merken. Um das zu erreichen, muß einiges beachtet werden: Das umschaltende Programm (eine Routine des Betriebssystems) darf sich natürlich nicht selbst wegschalten. Es muß daher in einem Speicherbereich liegen, der auch nach dem "Switching" erhalten bleibt. Dieser Speicherbereich, der allen "Banks" gemeinsam angehört, heißt "COMMON AREA". Weiter wird ein Anwenderprogramm, welches in einem "ausgeblendeten" Speicherblock liegt, einen "Stack" besitzen, auf dem Rückkehradressen und gegebenenfalls auch Variable gespeichert sind. Dieser Stack ist nach dem Ausblenden nicht mehr greifbar. Nach dem "Einblenden" des Blockes muß er natürlich wieder da sein. Es ist also ein "Stack Saving" erforderlich, was normalerweise folgendermaßen geschieht:

Der Stackpointer für das ausgeblendete Programm wird gespeichert. Für die Routinen im eingeblendeten Block wird ein neuer Stack eingerichtet. Wenn diese beendet sind, ist auch der neue Stack "abgearbeitet". Dann kann der alte Stackpointer wieder hergestellt und zurückgeschaltet werden. Abbildung 3a zeigt, wie der Speicher beim CPC unter **BASIC** aussieht. Bei der ursprünglichen Konfiguration war nur ein **Kassettenlaufwerk** angeschlossen. Für seine "Versorgung" sind Routinen im "unteren System-ROM" vorgesehen. Schließt man ein **Diskettenlaufwerk** an, so braucht man eine größere Anzahl weiterer Routinen, die in einem "DOS" (Disk Operating System) zusammengefaßt sind. Amstrad hat

```
          +----------------------+-----------------+ FFFF
          !      VIDEO-RAM       ! AMSDOS-ROM      !
          !                      ! (VDOS)          !
 .......  +----------------------+-----------------+ C000
 .        ! Syst.Bereich(Blocksw.)!
 .        ! BIOS Tab. Sprungleiste!
 .        +----------------------+ AD00
 .        !                      !
 .        !      BDOS            !
 .        !                      !
 .        !                      !
 .        +----------------------+ 9F00
 .        !                      !
COMMON    !      CCP             !
AREA      !                      !
 .        +----------------------+ 9700
 .        !                      !
 .        !                      !
 .        !                      !
 .        !      TPA             !
 .......  !                      +-----------------+ 4000
          !                      !                 !
          !                      ! Betriebssyst.   !
   0100   +----------------------+ (unteres) ROM   !
          !      ZEROPAGE        !                 !
          +----------------------+-----------------+ 0000
```

Abb. 3b: Der Speicher des CPC unter CP/M 2.2

dafür ein eigenes ROM entwickelt (AMSDOS-ROM), welches bei Bedarf im Adreßbereich C000 bis FFFF (an Stelle des BASIC-ROMs) eingeblendet werden kann. Für ein VORTEX-Laufwerk gibt es das VDOS-ROM im gleichen Adreßbereich. Bei der Installation von CP/M auf dem CPC war nun – wie üblich – vom Hersteller ein BIOS zu erstellen. Der Einfachheit halber haben die Amstrad-Entwickler dabei das vorhandene AMSDOS und das "untere System-ROM" mitverwendet. Für die "BIOS-Sprungleiste" und die erforderlichen Tabellen war dann noch ein gewisser RAM-Bereich erforderlich, und als Ergebnis erhielt man – als Folge der eingesparten Programmierarbeit – die Speicheraufteilung von Abbildung 3b. Die Ursachen für die zu kleine TPA sind damit erläutert. Was kann man dagegen tun? Will man die TPA vergrößern, so braucht man auf jeden Fall zusätzliches RAM. Da der Adreßbereich des Prozessors ohnehin nicht ausreicht, muß man mit einem weiteren Block- beziehungsweise Bankswitching arbeiten, um das Zusatz-RAM bei Bedarf ein- und auszublenden. Man kann nun die CP/M-Version 2.2 beibehalten und das BIOS so erweitern, daß von dort das zusätzliche Bankswitching durchgeführt werden kann. Eine solche Methode wurde früher bereits bei verschiedenen CP/M-Rechnern angewandt. Für den CPC gibt es die Speichererweiterung der Firma VORTEX, die nach diesem Prinzip funktioniert. Man erhält hardwaremäßig Zusatz-RAM und softwaremäßig ein entsprechendes BIOS. Damit ergibt sich eine TPA von etwa 60 kByte, auf der praktisch alle CP/M-Programme (unter der Version 2.2 laufen).

Man kann aber auch – wenn die hardwaremäßigen Voraussetzungen vorhanden sind – die bereits 1982 entwickelte neue Version 3 des CP/M (CP/M Plus) verwenden. Bei ihr ist eine TPA-Erweiterung mit der Methode des Bankswitching im Systemprogramm vorgesehen.

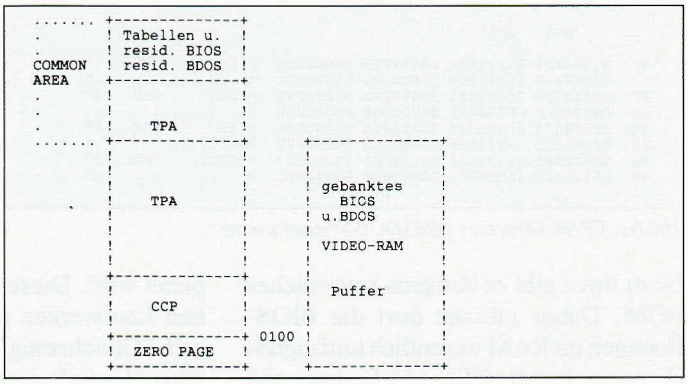


Abb.4: Allgemeine Speicheraufteilung bei CP/M Plus

Abbildung 4 zeigt die allgemeine Speicheraufteilung bei CP/M Plus. Bank 1 ist die TPA- oder Anwenderbank. Sie enthält die Zeropage und eine TPA von mindestens 60 kByte, in welche die transienten Programme geladen werden. Darüber befindet sich nicht mehr das ganze Betriebssystem (wie bei der Version 2.2), sondern nur noch ein kleiner Teil vom BIOS und BDOS. Der CCP ist nicht mehr unterhalb des BDOS, sondern wird wie ein transientes Programm ab 0100 in die TPA geladen und dort dann vom Anwenderprogramm überschrieben. (An seine Stelle ist unterhalb des BDOS ein kleiner Programmlader getreten.)

Die in Bank 1 befindlichen Teile des Betriebssystems sind resident und befinden sich daher in der COMMON AREA. Im BDOS-Teil sind die Einsprünge für die BDOS-Funktionen, im BIOS-Teil die Sprungleiste, die Disk-Parameter und die Funktionen zur Ansteuerung der Hardwarelogik für das Bankswitching untergebracht. Die Common Area muß natürlich mindestens so groß sein, daß all diese residenten Programme darin Platz haben. Meist ist sie noch etwas größer, so daß sich auch noch ein Teil der TPA darin befindet, was weiter nicht stört (übliche Größe beim CPC und Joyce ist 16 kByte).

Wenn nun vom Anwenderprogramm eine BDOS-Funktion aufgerufen wird, so erfolgt normalerweise vom residenten Teil des Systems (in der Common Area) eine Umschaltung auf Bank 0 (Systembank). Hier sind alle restlichen Teile des Systems (BDOS und BIOS vorhanden und das Anwenderprogramm kann entsprechend "bedient" werden. Danach wird wieder auf Bank 1 zurückgeschaltet. Dieses Konzept des Bankswitching mit einer Common Adress am oberen Ende des Speichers ist kennzeichnend für CP/M Plus. Es können noch weitere Speicherbänke vorhanden sein (insgesamt sechzehn Stück), die dann zum Beispiel als RAM-Disk verwendet werden (Joyce). Meist wird in einer dieser Bänke der CCP untergebracht. Wenn dieser bei einem Warmboot dann neu geladen wird, so geht das von der Speicherbank aus natürlich viel schneller als von der Diskette. Abbildung 10 zeigt die Speicheraufteilung von CP/M Plus auf dem CPC 6128. Dieser Rechner hat bereits ein RAM von zweimal 64 kByte und eine hardwaremäßige Bankumschaltlogik, so daß hier CP/M Plus installiert werden kann. Das Umschaltkonzept von CP/M Plus ist anders als das AMSTRAD-Konzept bei BASIC und CP/M 2.2 (vergleiche dazu Abbildung 3b und 4). Für CP/M Plus ist eine Common-Adress im Bereich C000 bis FFFF erforderlich, so daß das AMSDOS-ROM hier nicht verwendet werden kann. Daher mußten die AMSTRAD-Programmierer neue Disketten-Routinen für CP/M Plus erstellen die im RAM der Bank 0 untergebracht sind. Für den Sachkundigen hat das den großen Vorteil, daß er BIOS-"Patches" bei Bedarf leicht ausführen kann. Die Systemroutinen im unteren ROM (0000 bis 3FFF werden auch unter CP/M Plus verwendet, denn von der Systembank aus kann dieses ROM jederzeit ein- und ausgeblendet werden.

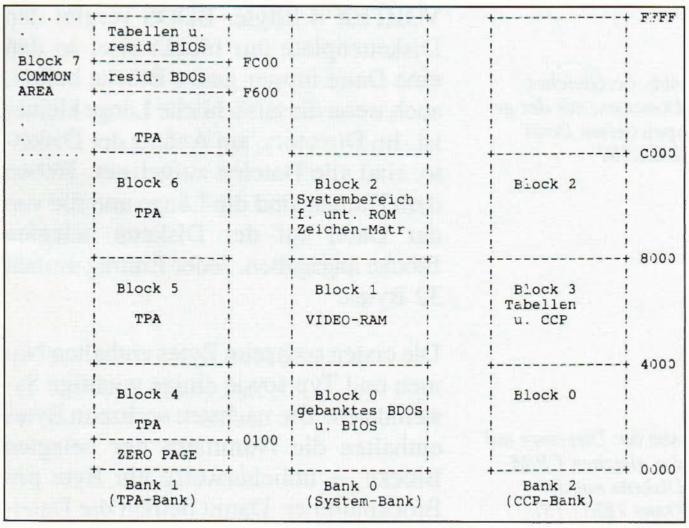


Abb.5: Der Speicher des CPC 6128 unter CP/M Plus. Anstelle von Block 0 kann das "untere" System-Rom eingeschaltet werden

```
        T=2,  S=1

 00  00504950 20202020 20434F4D 0000003A  *.PIP     COM...:*
 10  02030405 06070809 00000000 00000000  *................*
 20  00444454 20202020 20434F4D 00000026  *.DDT     COM...&*
 30  0A0B0C0D 0E000000 00000000 00000000  *................*
 40  00454420 20202020 20434F4D 00000034  *.ED      COM...4*
 50  0F101112 13141500 00000000 00000000  *................*
 60  00535441 54202020 20434F4D 00000029  *.STAT    COM...)*
 70  16171819 1A1B0000 00000000 00000000  *................*
```

Abb.6a: CP/M-Directory (AMSDOS-Systemformat)

```
        T=2,  S=1

 00  00504950 20202020 20434F4D 0000003A  *.PIP     COM...:*
 10  02030405 06070809 00000000 00000000  *................*
 20  00444454 20202020 20434F4D 00000026  *.DDT     COM...&*
 30  0A0B0C0D 0E000000 00000000 00000000  *................*
 40  E5454420 20202020 20434F4D 00000034  *eED      COM...4*
 50  0F101112 13141500 00000000 00000000  *................*
 60  00535441 54202020 20434F4D 00000029  *.STAT    COM...)*
 70  16171819 1A1B0000 00000000 00000000  *................*
```

Abb.6b: CP/M-Directory. Die Datei ED.COM wurde mit ERA gelöscht

Beim Joyce gibt es übrigens kein solches ROM. Daher müssen dort die BIOS-Routinen im RAM wesentlich umfangreicher sein. Für die CPC-464-Besitzer gibt es eine Speichererweiterung von der Firma dk'tronics, welche die für CP/M Plus erforderliche Hardwarelogik besitzt. Das gilt aber nicht für die VORTEX-Erweiterung.

Die Disketten unter CP/M

Wir haben gesehen, daß sich CCP und BDOS sowie auch Teile des BIOS im RAM-Speicher befinden. Sie werden von einer CP/M-Systemdiskette dorthin geladen. Wie das im einzelnen geschieht, werden wir noch erfahren. Zunächst wollen wir uns allgemein mit den Disketten unter CP/M etwas näher befassen. CP/M ist ein "single user"- und "single task"-Betriebssystem. Das heißt, zu gleicher Zeit kann nur ein Benutzer an **einer** Aufgabe (praktisch also an **einem** Programm) arbeiten. Es ist auch nur **ein** Laufwerk "angemeldet", auf welches CP/M zugreift. Soll auf ein anderes Laufwerk zugegriffen werden, so muß es vom Benutzer (eventuell auch durch das laufende Programm) angemeldet werden. Zu jedem Laufwerk gehören Tabellen – vor allem die DPBs (Disk-Parameter-Blöcke) die sich im BIOS befinden. Damit stellt sich das Betriebssystem bei der Anmeldung eines Laufwerks auf das Diskettenformat ein, welches dort verarbeitet wird. Dieses kann auf den einzelnen Laufwerken durchaus verschieden sein. Gleichzeitig wird bei der Anmeldung "Login" eines Laufwerks noch eine "Belegungstabelle" der gesamten Diskette und eine Prüfsumme über das Inhaltsverzeichnis "Directory" erstellt. Das sind Schutzmaßnahmen gegen einen versehentlichen Diskettenwechsel. Würde dieser nämlich unkontrolliert stattfinden, so würde – da das Directory der neuen Diskette mit dem der alten nicht übereinstimmt – ein Schreibzugriff nach dem alten Muster im Directory ausgeführt. Dabei würden sicherlich vorhandene Dateien zerstört werden. Aus diesem Grund wird die Directory-Prüfsumme immer vorher kontrolliert. Erkennt CP/M einen Diskettenwechsel, so sperrt es bei der Version 2.2 den Schreibzugriff.

Daher müssen – **bei CP/M 2.2** – Diskettenwechsel durch ein neues Login dem System immer mitgeteilt werden. Das kann sich gelegentlich sehr unangenehm auswirken. Man hat zum Beispiel eine mühsam erstellte, größere Datei im Speicher und stellt dann fest, daß der Platz auf der Diskette für eine Speicherung nicht mehr ausreicht. Sowohl bei einem "heimlichen" Diskettenwechsel als auch bei einem neuen login wird das laufende Programm mit einem Warmstart abgebrochen, und die Daten im Speicher sind normalerweise futsch, wenn man sie nicht durch besondere Tricks noch retten kann.

```
        T=2,  S=1
 00  00504950 20202020 20434F4D 0000003A  *.PIP     COM...:*
 10  02030405 06070809 00000000 00000000  *................*
 20  00444454 20202020 20434F4D 00000026  *.DDT     COM...&*
 30  0A0B0C0D 0E000000 00000000 00000000  *................*
 40  00544553 54202020 20545354 00000018  *.TEST    TST....*
 50  0F101100 00000000 00000000 00000000  *................*
 60  00535441 54202020 20434F4D 00000029  *.STAT    COM...)*
 70  16171819 1A1B0000 00000000 00000000  *................*
```

Abb. 6c:Gleiches Directory, mit der gespeicherten Datei TEST.TST

```
        T=2,  S=1

 00  00504950 20202020 20434F4D 0000003A  *.PIP     COM...:*
 10  02030405 06070809 00000000 00000000  *................*
 20  00444454 20202020 20434F4D 00000026  *.DDT     COM...&*
 30  0A0B0C0D 0E000000 00000000 00000000  *................*
 40  00544553 54202020 20545354 00000018  *.TEST    TST....*
 50  0F101100 00000000 00000000 00000000  *................*
 60  00535441 54202020 20434F4D 00000029  *.STAT    COM...)*
 70  16171819 1A1B0000 00000000 00000000  *................*
        T=2,  S=2
 00  00544553 54312020 20545354 00000030  *.TEST1   TST...0*
 10  12131415 1C1D0000 00000000 00000000  *................*
 20  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
 30  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
 40  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
 50  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
 60  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
 70  E5E5E5E5 E5E5E5E5 E5E5E5E5 E5E5E5E5  *eeeeeeeeeeeeeeee*
```

Abb.6d: Directory auf der gleichen CP/M-Diskette mit der Datei TEST1.TST

Hier hilft CP/M Plus. Bei dieser Version führt der vom System erkannte Diskettenwechsel automatisch zu einem neuen Login, und alles ist gerettet. Eine Diskette besteht **physikalisch** aus **Zylindern** und **Sektoren**. Die Sektoren können sich nur auf der Vorderseite des Zylinders oder zusätzlich auch auf seiner Rückseite befinden. Die Größe eines "physikalischen" Sektors ist bei den einzelnen Formaten verschieden. Beim CPC und Joyce beträgt sie immer 512 Bytes. **Logisch** – aus der Sicht von CP/M – besteht die Diskette aus **Spuren** und **Sektoren**. Wenn sich auch auf der Rückseite Sektoren befinden, dann kann eine **logische Spur** auf der Vorderseite beginnen und sich auf der Rückseite fortsetzen. Sie kann aber auch nur eine Seite eines Zylinders umfassen. Auf der anderen Seite ist dann eine andere Spur. Jede logische Spur enthält eine bestimmte Anzahl von **logischen** (oder auch **CP/M-**) Sektoren mit einer Größe von 128 Bytes. Im Verkehr mit einem Programm sieht **BDOS** nun die Diskette noch etwas anders, nämlich als ein Speichermedium, auf dem sich zusammenhängende Informationen in Form von Dateien ("Files") befinden. Jede Datei hat einen Namen, unter dem sie von CP/M erfaßt wird. (Bis zu 8 Zeichen und zusätzlich noch wahlweise bis zu 3 Zeichen für einen Dateityp.) Für die Verwaltung der Dateien teilt BDOS die Diskette in Blöcke (Allocation Blocks) auf. Die Länge eines solchen Blocks ist bei den einzelnen Formaten verschieden. Bei AMSDOS 1 kByte, bei VORTEX 4 kByte. BDOS vergibt den Diskettenplatz nur blockweise, so daß eine Datei immer ganze Blöcke belegt, auch wenn die tatsächliche Länge kleiner ist. Im Directory, am Anfang der Diskette, sind alle Dateien aufgelistet. Neben dem Namen sind die Länge und die von der Datei auf der Diskette belegten Blöcke angegeben. Jeder Eintrag umfaßt 32 Bytes.

Die ersten sechzehn Bytes enthalten Namen und Typ sowie einige wichtige Systemdaten. Die nächsten sechzehn Bytes enthalten die Nummern der belegten Blöcke – üblicherweise ein Byte pro Blocknummer. Damit beträgt die Datei-

länge pro Eintrag bei AMSDOS 16 x 1 = 16 kBytes, bei VORTEX 16 x 4 = 64 kBytes. Ist die Datei länger, so werden weitere Einträge unter demselben Dateinamen benötigt. Eine entsprechende Kennzeichnung sagt dem System, daß es sich um die gleiche Datei handelt.

Der Datentransfer zwischen Diskette und Anwenderprogramm erfolgt über bestimmte BDOS-Funktionen. Für diesen Transfer wird die Datei in einzelne Sätze "records" mit einer Länge von 128 Bytes aufgeteilt. Das entspricht der Länge eines CP/M-Sektors. Dem BDOS muß dann der Dateiname übergeben werden. Soll nicht die ganze Datei von Anfang an, also sequentiell, verarbeitet werden, so muß auch noch die gewünschte Satznummer genannt werden. Mit Hilfe der Directory-Einträge berechnet BDOS daraus die logische Spur und den logischen Sektor. Diese Werte werden an das BIOS weitergegeben, als Parameter über die BIOS-Sprungleiste. Der an die Hardware eng angepaßte Teil des BIOS muß daraus dann die "physikalischen" Werte ermitteln. Diese Daten werden dem FDC (Floppy-Disk-Controller) übergeben, und damit wird das Laufwerk angesteuert. Das Directory befindet sich am "logischen" Anfang der Diskette. Hier beginnt auch die Numerierung der Blöcke, das heißt, das Directory belegt immer Block 0. (Bei Bedarf noch weitere Blöcke.) Den "physikalischen" Anfang der Diskette bilden die "Systemspuren". Im ersten Sektor der ersten Spur (Spur 0) befindet sich ein "Bootprogramm" (von dem wir gleich noch mehr hören werden). Dann folgt allerdings nur bei der Version 2.2 das eigentliche Betriebssystem BIOS und BDOS. Insgesamt sind bei einem normalen CP/M-Rechner drei bis vier Spuren als Systemspuren belegt. Beim CPC ist es etwas günstiger, weil der größte Teil des BIOS bereits im ROM vorhanden ist. Für die vorhandenen zwei Systemspuren ergibt sich folgende Belegung (für die Version 2.2): *Spur 0, Sektor 1 = Bootprogramm. Sektor 2 = verkürztes BIOS. Sektor 3 bis 7 = unbenutzt. Spur 0, Sektor 8 bis 9, und Spur 1, Sektor 1 bis 2 = CPC. Spur 1 Sektor 3 bis 9 = BDOS.* Bei der Version 3 (CP/M Plus) ist das Betriebssystem so umfangreich geworden, daß es sich nicht mehr in den Systemspuren unterbringen läßt. Es befindet sich in einer eigenen Datei (Systemfile) im Datenteil der Diskette und ist daher auch – wie alle anderen Diskettendateien – im Directory aufgeführt. (Beim CPC 6128 hat das Systemfile die Bezeichnung **C10CPM3.EMS** "EMS"-Datei). Auf der Spur 0 befindet sich jetzt nur das Bootprogramm (Version 3 im Sektor 1). Der Rest der Systemspuren ist leer, was eigentlich eine Verschwendung von Diskettenplatz bedeutet. Bei MS-DOS ist das übrigens besser organisiert.

Das Booten von CP/M

Wir wissen jetzt, wo sich das Betriebssystem auf der Diskette befindet, und müssen nun noch erläutern, wie es von dort in den Speicher geladen wird. Nach dem Einschalten des CPC ist bekanntlich das BASIC-ROM aktiviert. Unter BASIC kann man Maschinenprogramme im RAM oder in einem ROM als sogenannte RSXs (Resident System eXtensions) aufrufen. Ein solches RSX ist auch CPM – eine Routine im AMSDOS-ROM. Diese ROM-Routine ist ein sogenannter **"Bootstrap"** . Er liest von einer Systemdiskette den ersten Sektor der ersten Spur in das RAM. Dieser Sektor enthält ein sogenanntes **"Bootprogramm"**, welches anschließend gestartet wird. Das Bootprogramm lädt dann BIOS und BDOS in das RAM, bei der Version 2.2 aus den Systemspuren, bei der Version 3 (CP/M Plus) als "EMS-Datei". Dann wird die BIOS-Routine "Coldboot" gestartet, welche Tabellen, Sprungleisten und vieles andere initialisiert. Zu guter Letzt folgt dann "Warmboot", wodurch der CCP geladen wird und das System sich mit dem "Prompt" melden kann.

Besitzt der Rechner kein BASIC-ROM – wie zum Beispiel der JOYCE – so hat er lediglich ein kleines "Bootstrap-ROM". Diese Bootstrap-Routine wird beim Einschalten automatisch (oder durch eine bestimmte Tastenkombination) gestartet, und dann läuft alles andere wie beschrieben.

Den gesamten Rechnerstart nennt man dann "Booten" oder auch "Coldboot" – im Gegensatz zum (darin enthaltenen) "Warmboot". Was aber haben die "Boots", also die Stiefel, dabei zu suchen? Erfunden wurden diese Bezeichnungen von den Pionieren der Computertechnik, die dabei an eine alte Redensart aus der Cowboyzeit angeknüpft haben. Die Stiefel hatten damals große Schlaufen, damit man sie besser anziehen konnte. Das waren die "Bootstraps". Ein "Tausendsassa" brachte es nun sogar fertig, sich an seinen eigenen Bootstraps in die Höhe zu ziehen. Für unsere Pioniere – die das Staunen noch nicht verlernt hatten – war der Start eines Rechners ein durchaus vergleichbares Kunststück. Damit wollen wir unseren Überblick über CP/M beenden. Einiges über die praktische Handhabung von CP/M-Kommandos finden Sie in dem Beitrag "Die residenten und transienten Kommandos von CP/M". Auf die angeführte Fachliteratur sei nochmals hingewiesen.

Sie sollten sich übrigens unbedingt eine CP/M-Systemdiskette mit einem Diskettenmonitor ansehen. Monitor-Programme sind mehrfach schon in der PCI veröffentlicht worden. Es sind allerdings alles BASIC-Versionen, die nicht unter CP/M laufen. Sie können damit aber ohne weiteres auch eine CP/M-Diskette untersuchen. Ein ausgezeichneter Disk-Monitor, der unter CP/M 2.2 und CP/M Plus läuft, ist das Programm **DU.COM**, welches Sie als Public-Domain-Software sehr preisgünstig erhalten. (Siehe Testbericht in PCI 11/88, S. 54/55.)

CP/M Intern II

Die residenten und transienten Commands von CP/M

In diesem Beitrag finden Sie Hinweise für den praktischen Umgang mit CP/M auf dem CPC. Und zwar für beide dort laufende Versionen – CP/M 2.2 und CP/M Plus. Er ist vor allem für den CP/M-Anfänger bestimmt. Aber das eine oder andere ist auch für den Fortgeschrittenen interessant.

Wir werden gelegentlich diesen Aufsatz als CP/M-Beitrag zitieren. Natürlich können wir nicht alle CP/M-Commands ausführlich besprechen. Näheres müssen Sie in der Fachliteratur nachlesen. Eine geeignete Auswahl – besonders auch für die praktische Arbeit – finden Sie am Schluß dieses Beitrags.

Da wir uns hier vorwiegend mit der Praxis beschäftigen wollen, so starten Sie gleich einmal CP/M 2.2 und – wenn es Ihre Hardware erlaubt – auch CP/M

```
A> DDT
DDT VERS 2.2
- F0100,01FF,00
- S0100
0100 00  11
0101 00  0B
0102 00  01
0103 00  0E
0104 00  09
0105 00  CD
0106 00  05
0107 00
0108 00  C3
0109 00
010A 00
010B 00  0C
010C 00  47
010D 00  75
010E 00  74
010F 00  65
0110 00  6E
0111 00  20
0112 00  4D
0113 00  6F
0114 00  72
0115 00  67
0116 00  65
0117 00  6E
0118 00  24
0119 00  .
- ^C
A> SAVE 1 TEXTAUS.COM
```

Abb.7: Protokoll der "DDT-Sitzung" unter CP/M 2.2

Plus jeweils mit der entsprechenden Systemdiskette. Beobachten Sie dabei den im "CP/M-Beitrag" beschriebenen Bootvorgang. Es finden zwei Ladevorgänge hintereinander statt:

1. Laden des Bootprogramms
2. Laden der Systemroutinen

Bei 2.2 erfolgt beides aus den Systemspuren und so schnell hintereinander, daß Sie es nicht voneinander trennen können. Bei CP/M Plus aber können Sie auseinanderhalten. Zunächst das Laden des Bootprogramms von der Systemspur 2 und dann das folgende Laden der EMS-Datei über das Directory (das dauert nämlich etwas länger). In jedem Falle meldet sich schließlich der CCP mit dem angemeldeten Laufwerk A (Bootlaufwerk) und dem Prompt " > ". Der Cursor zeigt, daß eine Eingabe erwartet wird. *In PCI 4/88, S.75/76, mit Ergänzung im Leserbrief, PCI 8/88, S.9, ist beschrieben, wie Sie sich einen "blinkenden" Cursor erzeugen können.*

ALLGEMEINES ÜBER CP/M

Geben Sie nun irgendeinen Text ein. **Jede Eingabe – von der in Zukunft die Rede ist – muß durch die Betätigung der ENTER- oder RETURN-Taste abgeschlossen werden.** Je nach Länge Ihrer Eingabe wird der CCP die ganze oder einen Teil Ihrer Eingabe mit einem Fragezeichen wiederholen, weil er Sie nicht verstanden hat. Er erwartet natürlich die Eingabe eines **residenten** oder **transienten** Kommandos.

Überzeugen Sie sich auch davon, daß Sie für die Eingabe Groß- oder Kleinschreibung wählen können; der CCP wandelt alles in Großbuchstaben um. Natürlich gilt das nur für Buchstaben. Bei allen anderen Zeichen ist zwischen Eingabe ohne oder mit SHIFT-Taste wohl zu unterscheiden. Probieren Sie auch die Texteingabe, nachdem Sie **vorher CTRL-P** eingegeben haben. Jetzt wird alles was auf dem Bildschirm steht – Ihre Eingabe und die Ausgabe des CCP – vom Drucker "protokolliert". Natürlich muß dieser eingeschaltet und "online" sein, sonst akzeptiert CP/M nichts mehr, bevor das erfolgt ist. Ausgeschaltet wird die Druckerausgabe durch ein erneut eingegebenes CTRL-P.

Beim CP/M Plus ertönt bei der ersten CTRL-P Eingabe ein "Beep". Dann geben Sie noch ein CTRL-C ein und erzeugen damit ein "Warmboot". Beim 2.2 erkennen Sie das Nachladen von CCP und BDOS von der Diskette. Beim CP/M Plus erfolgt das Nachladen von CCP aus einer Speicherbank, was schnell und lautlos vonstatten geht. Die Bedeutung von CTRL-C für das "Login" einer Diskette werden wir später noch behandeln. Wenn Sie ein zweites Laufwerk mit eingelegter Diskette angeschlossen haben, so können Sie darauf durch Eingabe von "B:" umschalten. Ist keine Diskette eingelegt, so bekommen Sie eine ERROR-Meldung. Nach Einlegen einer Diskette können Sie "RETRY" machen, und alles ist in Ordnung. Das Laufwerk B ist "eingelogged" und damit für das Betriebssystem angemeldet. Durch Eingabe von "A:" wird wieder Laufwerk A angewählt. Wenn Sie kein zweites Laufwerk angeschlossen haben, bekommen Sie auch eine ERROR-Meldung, aus der Sie aber – trotz aller Anstrengungen – kaum herauskommen werden (Versuchen Sie es!). Es bleibt schließlich nur der "Totale RESET" mit dem Dreifingergriff (ESC-SHIFT-CONTROL). Denken Sie daran, wenn Sie wichtige Daten im RAM haben, die noch nicht abgespeichert sind.

CP/M 2.2 RESIDENTE COMMANDS

Jetzt wollen wir uns mit den residenten Commands von CP/M 2.2 beschäftigen. Sie lauten *DIR, ERA, REN, TYPE, USER* und *SAVE.* DIR kennen Sie natürlich. Die Ausgabe der Dateinamen erfolgt genau in der Reihenfolge, wie die Dateien im Directory auf der Diskette gespeichert sind. *Mit einem Diskettenmonitor – siehe dazu die Ausführungen im CP/M-Beitrag – können Sie sich davon überzeugen.* Sie können DIR auch in Verbindung mit "Wildcards" einsetzen. Was ist das? Nehmen wir an, Sie wollen sich alle PASCAL-Dateien anzeigen lassen. Da diese vom Dateityp ".PAS" sind so geben Sie ein **DIR *.PAS.** Nehmen wir weiter an, Sie haben Dateien von Jahresbilanzen mit den Namen BILANZ86, BILANZ87 und so weiter. Zusätzlich haben diese Dateien noch den Typ .EIG oder .FRD, je nachdem, ob es sich um Ihre eigenen oder fremde Bilanzen handelt. Mit der Eingabe **DIR BILANZ??.*** bekommen Sie **alle**, mit **DIR BILANZ??.EIG** nur die **eigenen** Bilanzdateien angezeigt. Der Stern steht also für eine beliebige und beliebig lange Bezeichnung, das Fragezeichen genau für ein beliebiges (erlaubtes Zeichen).

Zur Erinnerung: der Dateiname kann bis zu acht Zeichen, der Typ bis zu drei Zeichen lang sein. Als Zeichen sollten Sie Zahlen oder Buchstaben, möglichst keine Sonderzeichen und keine deutschen Umlaute verwenden. ERA wird Ihnen auch bekannt sein. Es "löscht" den Eintrag eines Dateinamens im Directory. Auch ERA können Sie in Verbindung mit "Wildcards" einsetzen. Seien Sie dabei vorsichtig, denn der CCP ist schnell bei der Hand. Lediglich, wenn Sie mit **ERA *.*** angeben, daß **alle** Dateien gelöscht werden sollen, fragt er zur Vorsicht noch einmal nach. An sich geschieht beim Löschen mit ERA noch nichts Schlimmes. Es wird lediglich das erste Byte im Directory-Eintrag, welches normalerweise 00 ist (siehe aber später unter USER), mit

```
0100    11 0b 01            LD DE,010B      ;Lade Text
0103    0E 09               LD C,09         ;Aufruf
0105    CD 05 00            CALL 0005       ;BDOS-Funktion 9
0108    C3 00 00            JP 0000         ;Warmboot
010B    0C                                  ;"Bildschirmlöschen"
010C    47 75 74 65 6E 20                   ;"Guten "
0112    4D 6F 72 67 65 6E 24                ;"Morgen$"
```

Listing 1a. TEXTAUS.COM

```
0100  11 0B 01 0E 09 CD 05 00 C3 00 00 0C 47 75 74 65
0110  6E 20 4D 6F 72 67 65 6E 24 00 00 00 00 00 00 00
```

Listing 1b: Hexdump von TEXTAUS.COM

E5 überschrieben. Der gesamte Eintrag – insbesondere die von der Datei belegten Blöcke – bleibt zunächst erhalten. Wenn weiter nichts geschehen ist, so können Sie eine versehentlich gelöschte Datei wieder retten. Wurde aber inzwischen eine andere Datei gespeichert, so ist der Platz der gelöschten Datei – oder zumindest ein Teil davon – anderweitig vergeben und überschrieben. Sehen Sie sich diese Vorgänge unbedingt mit einem Diskettenmonitor an, Abbildung 6a...d zeigt die Untersuchung eines Beispiels mit DU.COM. In Abbildung 6a sehen Sie das ursprüngliche Directory. Dann wurde die Datei ED.COM mit ERA gelöscht. Abbildung 6b zeigt das Ergebnis. Achten Sie auf die erhaltene Blockbelegung. (Will man die Datei jetzt wieder retten, so kann man mit DU das "E5" wieder mit "00" überschreiben.) Man kann auch ein Hilfsprogramm UNERA.COM benutzen. (Es steht auf der gleichen Public-Domain-Diskette wie DU.COM.) Jetzt wurde die Datei TEST.TST gespeichert. Für das Experiment wird dazu einfach ein Teil des RAMs – von 0100 bis 0CFF – das sind zwölf Pages = 3 Blöcke – mit dem Command SAVE (siehe weiter unten) abgespeichert. In Abbildung 6c können Sie sehen, wie das Directory jetzt aussieht. Die Datei ED.COM ist verschwunden, die von ihr belegten Blöcke tauchen nicht mehr auf. Statt dessen belegt TEST.TST die Blöcke 0F 10 11. Auf den Blöcken 12 13 14 15 ist natürlich der Rest von ED.COM noch vorhanden (Sie können sich mit dem Monitor überzeugen). Aber man sieht davon nichts, denn diese Blöcke sind CP/M als frei gemeldet und stehen bei Bedarf zur Verfügung. Wenn jetzt die Datei TEST1.TST gespeichert wird (als RAM von 0100 bis 18FF, das sind 24 Pages = 6 Blöcke), dann belegt TEST1.TST die restlichen ehemaligen Blöcke von ED.COM – 12 13 14 15 – und weil das nicht reicht, zusätzlich noch 1C und 1D (Abbildung 6d). Damit ist dann ED.COM endgültig überschrieben. Das Kommando REN wird Ihnen auch bekannt sein. Sie ändern damit den Namen einer vorhandenen Datei, REN neuer Name = alter Name, zum Beispiel **REN NEU.DAT ALT.DAT** .

Mit TYPE geben Sie den Inhalt einer ASCII-(Text-)Datei auf dem Bildschirm aus: **TYPE DATA.TXT**. Haben Sie vorher CTRL-P eingegeben, so erfolgt die Ausgabe auch auf dem Drucker. Wenn Sie die Ausgabe anhalten wollen, so geben Sie CTRL-S ein. Durch Eingabe von CTRL-Q (oder irgendeinem anderen Zeichen) wird die Ausgabe fortgesetzt.

TYPE sollte **nur** bei Textdateien verwendet werden. Die Binärcodes anderer Dateien ergeben meist keine darstellbaren Zeichen. Probieren Sie TYPE einmal bei einer anderen Datei (.COM oder auch .BAS). Sie werden erstaunt sein, was sich da auf Ihrem Bildschirm tut. (Den Drucker schalten Sie besser nicht ein!) Wenn Sie "Glück" haben, erleben Sie vielleicht sogar einen "Systemabsturz" (mit "nichts geht mehr").

USER werden wir ausführlich bei CP/M Plus besprechen. SAVE dient dazu, einen Teil des RAMs – genauer gesagt der TPT – als Datei auf der Diskette zu speichern. Es fängt immer bei 0100 an. Die Länge des Bereichs wird in "Pages" angegeben (Pageanzahl **dezimal** 1 Page = 256 kBytes, das heißt, 4 Pages = 1 KByte).

SAVE 12 TEST.TST speichert 12 = 0C pages – also den Bereich von 0100 bis 0CFf – als Datei TEST.TST auf der Diskette. Benötigt wird SAVE vor allem, wenn "Patches" mit DDT ausgeführt wurden. Wir werden das gleich sehen.

CP/M 2.2 TRANSIENTE COMMANDS

Nun kommen wir zu den **transienten** Commands von CP/M 2.2. Diese bedeuten bekanntlich den Aufruf eines Programms mit dem entsprechenden Namen (also einer Datei vom Typ **.COM**). Dazu müssen sie natürlich auf der Diskette im angemeldeten Laufwerk vorhanden sein, sonst gibt der CPC seine stereotype ERROR-Meldung (Wiederholung des Namens mit einem Fragezeichen). Auf den Systemdisketten sind alle diese Programme vorhanden. Wir können hier nur einige – und auch keineswegs erschöpfend – besprechen. PIP.COM ist das universelle Kopierprogramm. Es setzt das Vorhandensein von **zwei** Laufwerken voraus. Unter CP/M 2.2 sind das zwei physikalische Laufwerke, und daher spielt PIP hier für die Mehrzahl der CPC-Benutzer keine große Rolle. Unter CP/M Plus ist das anders. Wir kommen später darauf zurück.

STAT.COM ist ein wichtiges Programm unter CP/M 2.2, mit dem einfachen Aufruf STAT wird der freie Speicherplatz auf der Diskette (im angemeldeten Laufwerk) angegeben. Wollen Sie wissen, wieviel Platz eine bestimmte Datei belegt, so geben Sie ein: **STAT Name.Typ**.

Wollen Sie die Platzbelegung von **allen** Dateien wissen, so geben Sie ein: **STAT *.***

Mit CTRL-S können Sie die Ausgabe – wie oben beschrieben – anhalten und durch vorherige Eingabe von CTRL-P gleichzeitig ausdrucken.

Die Platzhalter "*.*" sind die "Wildcards" für alle Dateien. Sie können auch die anderen beschriebenen "Wildcards" verwenden.

DDT.COM (kein Insektenvertilger, sondern *Dynamic Debugging Tool*) ist ein außerordentlich nützliches Hilfsprogramm. Sie sollten in der Fachliteratur nachlesen, was man alles damit machen kann. Häufig wird es benutzt, um vorhandene Dateien abzuändern (also zu "patchen") oder um kleinere Hilfsroutinen zu erstellen. In dieser Eigenschaft begegnet es einem immer wieder in den Computerzeitschriften. Ebenso wird es benutzt, um eine vorhandene Datei – oder auch einen Speicherbereich – zu "dumpen", das heißt, den Inhalt hexadezimal und in ASCII auszugeben. (Wenn der Hexcode keinem ASCII-Zeichen entspricht, so wird ein Punkt ausgegeben.) Wir geben Ihnen zwei Beispiele. Es soll ein Programm erstellt werden, welches den Bildschirm löscht und dann den Text "Guten Morgen" ausgibt. Wir nennen es TEXTAUS.COM. Dann soll das Programm gepatcht werden, so daß es den Text "Guten Abend" ausgibt. Aus dem CP/M-Beitrag wissen Sie, daß Ein- und Ausgabeoperationen durch den Aufruf entsprechender BDOS-Funktionen erfolgen. Für unsere Zwecke gibt es die BDOS-Funktion Nr. 9, welche eine Kette von ASCII-Zeichen auf dem Bildschirm ausgibt, darunter können sich auch Steuerzeichen wie "Bildschirmlöschen" befinden. Am Schluß der Kette muß das Dollarzeichen stehen, damit BDOS das Ende erkennt. Im Register DE wird die

```
A> DDT TEXTAUS.COM
DDT VERS 2.2
NEXT  PC
0180 0100
- D0100,017F
0100 11 0B 01 0E 09 CD 05 00 C3 00 00 0C 47 75 74 65 ............Gute
0110 6E 20 4D 6F 72 67 65 6E 24 00 00 00 00 00 00 00 n Morgen$.......
0120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
- S0112
0112 4D  41
0113 6F  62
0114 72  65
0115 67  6E
0116 65  64
0117 6E  20
0118 24
0119 00  .
- ^C
A> SAVE 1 TEXTAUS.COM
```

Abb.8: Protokoll der "DDT-Sitzung" zum Patchen des TEXTAUS.COM

Adresse der Zeichenkette übergeben. Die Funktionsnummer wird immer im Register C übergeben und die Funktion über die Adresse 0005 in der Zeropage aufgerufen. Im Listing 1a ist unser kleines Programm dargestellt. Es ist der Assemblercode angegeben und der vom Assemblerprogramm erstellte Hexadezimalcode. Auch wenn Sie kein "Assemblerspezialist" sind, werden Sie den besprochenen Ablauf erkennen. Das erste Zeichen in unserer Zeichenkette ist der Steuercode 0C CTRL L für das Löschen des Bildschirms (unter CP/M 2.2). Das Programm endet – wie üblich – mit einem Sprung zur Adresse 0000 (Warmboot). Listing 1b stellt dann einen "Hexdump" unseres Programms dar, wie er beispielsweise in einer Zeitschrift veröffentlicht würde. Die Eingabe erfolgt mit DDT. Ein "Protokoll" dieser Operation ist in Abbildung 7 dargestellt. Alle Ihre Eingaben sind **fett** gedruckt. Vergessen Sie nicht die RETURN-(oder ENTER-) Taste nach jeder Eingabe. Durch Eingabe von **DDT** wird das Programm aufgerufen. Es meldet sich mit Ausgabe der Versionsnummer und mit seinem "Prompt", einem waagrechten Strich ("-"). Der Cursor zeigt wieder, daß eine Eingabe erwartet wird. Es gibt mehr als ein Dutzend Commands für DDT, wie Sie in der Fachliteratur nachlesen können.

Wir benutzen zuerst das Command F (fill). Es lautet **Fadresse1,adresse2,wert**. Damit füllen Sie den RAM-Speicher von adresse1 bis adresse2 mit wert, hier also von 0100 bis 01FF mit 00. Das ist eine kleine – nicht unbedingt erforderliche – "Schönheitsoperation". Wir müssen nämlich später ein Page abspeichern, und darin soll – außer dem Programm – nichts anderes stehen.

Zur Eingabe des Programms benutzen wir das Command S (store). Wir müssen eingeben: **S adresse** also **S0100**. DDT gibt dann die Adresse aus und dahinter das Byte, welches sich jetzt in dieser Adresse befindet. Das ist hier 00, weil wir dies mit "F" ja dorthin gebracht haben. Dahinter können Sie nun den neuen Wert eingeben, der an der Adresse gespeichert werden soll. Das ist hier **11** entsprechend Listing 1b. Nach dem RETURN zeigt DDT die nächste Adresse an, und Sie können mit der Eingabe fortfahren. Wenn sich der angezeigte Wert im RAM nicht ändern soll, so geben Sie nichts ein, sondern bestätigen nur mit der RETURN-Taste. Wenn alles eingegeben ist, beenden Sie den Modus für das S-Command mit der Eingabe eines Punktes. Dann meldet sich DDT wieder mit seinem Prompt und zeigt dadurch an, daß es ein weiteres Command erwartet. Sie können sich zum Beispiel das eingegebene Programm als Hexdump – entsprechend Listing 1b – ansehen.

```
A> SID
CP/M 3 SID - Version 3.0
# F0100,01FF,00
# S0100
0100 00  11
0101 00  0B
0102 00  01
0103 00  0E
0104 00  09
0105 00  CD
0106 00  05
0107 00
0108 00  C3
0109 00
010A 00
010B 00  1B
010C 00  48
010D 00  1B
010E 00  45
010F 00  47
0110 00  75
0111 00  74
0112 00  65
0113 00  6E
0114 00  20
0115 00  4D
0116 00  6F
0117 00  72
0118 00  67
0119 00  65
011A 00  6E
011B 00  24
011C 00  .
# W TEXTAUS.COM,0100,011B
0001h record(s) written.
# ^C
A>
```

Abb.9: Protokoll zur Eingabe des Programms TEXTAUS.COM unter CP/M Plus

Dazu benutzen Sie das Command Display. Es lautet **Dadresse1,adresse2** hier also **D0100,011F**. Dabei wird der Textteil des Programms auch in ASCII ausgegeben, so daß Sie es kontrollieren können. Sie können auch das Programm, welches ja zunächst nur im RAM steht, von DDT aus mit dem Command G (go) starten. Die Startadresse ist 0100, also müssen Sie eingeben **G0100**. Versuchen Sie es einmal. Grundsätzlich ist dies Verfahren aber nicht zu empfehlen – jedenfalls bei einem längeren Programm. Es könnte ja sein, daß durch einen Eingabefehler das Programm nicht ordnungsgemäß läuft, sondern sich "in einer Endlos-Schleife" aufhängt oder sogar einen Systemabsturz verursacht. In diesem Falle wären alle Ihre Eingaben "futsch", und Sie müßten neu anfangen. Der Abschluß des Programms DDT erfolgt durch Eingabe von CTRL-C (Warmboot). Sie können natürlich auch **G0000** eingeben, dann meldet sich wieder der CCP mit seinem Prompt. Jetzt muß das im RAM stehende Programm auf der Diskette gespeichert werden. Dazu dient das erwähnte residente Command SAVE. Es lautet **SAVE anzahl Dateiname.Typ**.

Dabei ist anzahl die Anzahl der Pages zu je 256 kBytes. Wenn das Programm von 0100 bis maximal 01FF geht – wie in unserem Falle –, so ergibt das ein Page. Wir geben ein **SAVE 1 TEXTAUS.COM**. Nach dem Abspeichern meldet sich wieder der CCP. (Vergleichen Sie alle Ihre Eingaben mit dem Protokoll in Abbildung 2.) Anschließend können Sie Ihr "transientes" Anwenderprogramm durch Eingabe von **TEXTAUS** starten. Der programmierte Text wird ausgegeben, und als Abschluß erfolgt Warmboot, und der CCP meldet sich wieder. Jetzt wollen wir das Programm für die Ausgabe von "Guten Abend" "patchen". (Sollten Sie bislang einen Fehler gemacht haben, so können Sie ihn nach dem gleichen Muster korrigieren.) Dazu rufen wir DDT mit dem Namen der zu patchenden Datei auf, also **DDT TEXTAUS.COM** . In Abbildung 8 sehen Sie ein Protokoll dieser "Patchsitzung", nach dem Sie sich richten können. DDT zeigt Ihnen an wie weit sich die zu patchende Datei im RAM erstreckt. Unter "NEXT" ist der erste **freie** Speicherplatz **nach** dem Dateiende angegeben (natürlich in Hexadezimal). Schreiben Sie sich diesen Wert immer auf, damit Sie später richtig "saven" können.

Unter PC (= Program Counter) steht der Wert 0100 als Zeichen, daß diese Adresse "im Zugriff" ist. Mit dem D-Command sehen wir uns unser Programm noch einmal in Hex und ASCII an. Wir erkennen, daß der zu patchende Text bei Adresse 0112 beginnt und geben ein **S0112**. Dann patchen wir den alten ASCII-Code mit dem neuen. Ein Buchstabe bleibt übrig, für den wir ein Leerzeichen einfügen (oder das Dollarzeichen vorrücken). Der Rest läuft ab wie oben beschrieben. Wenn Sie bei SAVE den Namen TEXTAUS.COM eingeben, so wird die neue Version gespeichert und dabei die alte überschrieben. Eine Backup-Datei (Typ *.BAK) wird **nicht** angelegt. Wollen Sie die alte Version behalten, so müssen Sie die neue Version unbedingt unter einem anderen Namen "saven" und später den Namen mit REN ändern.

Noch ein Hinweis, wie Sie die Anzahl Pages für SAVE ermitteln. Von dem Wert NEXT nehmen Sie die beiden ersten Ziffern als Hexzahl. Wenn NEXT mit 00 endet, muß 1 davon subtrahiert werden, andernfalls bleibt die Zahl unverändert. Der Wert den Sie noch in eine Dezimalzahl verwandeln müssen, ergibt dann die Anzahl pages. Beispiel: NEXT = 0C80 0C (unverändert gleich 12 Pages (Dezimal)

NEXT = 1D00 1D minus 1 gleich 1C gleich 28 Pages. Hexadezimales Rechnen sollten Sie etwas üben, denn DDT bietet nur eine einseitige Hilfe.
Mit **Hdwert1,dwert2** erhalten Sie Summe und Differenz der Dezimalzahlen dwert1 und dwert2 in Hex.
Weitere transiente Commands von CP/M 2.2 wollen wir nur kurz erwähnen: MOVCPM und SYSGEN sind Programme zur Veränderung der TPA-Größe, wenn ein neues BIOS erstellt wurde. Sie werden benötigt, wenn Sie eine VORTEX-Floppy installieren. ED ist ein zeilenorientiertes Textverarbeitungsprogramm aus der Steinzeit, das in der Fachliteratur immer noch beschrieben wird. Hüten Sie sich davor! Es soll Leute gegeben haben, die beim Arbeiten mit diesem Programm aus Verzweiflung den Monitor zertrümmerten. ASM ist ein Assembler für "8080 Code". Wir wollen jetzt mit Hilfe des SAVE-Commands noch ein Experiment machen, das Ihnen die Notwendigkeit zeigt, bei einem Diskettenwechsel die neue Diskette "anzumelden" (siehe "CP/M-Beitrag"). Stellen Sie sich vor, Sie haben Daten im RAM, welche auf der Diskette gespeichert werden sollen. Sie merken, daß auf der im Laufwerk befindlichen Diskette nicht mehr genügend Platz ist, und legen eine neue ein. Jetzt versuchen Sie, Ihre Daten zu "saven". Geben Sie als Beispiel ein: **SAVE 3 TEST.TST**. Ergebnis CP/M streikt mit einer ERROR-Meldung. Geben Sie CTRL-C ein und melden mit dem Warmboot die neue Diskette an. Jetzt können Sie anschließend speichern. Bei diesem Experiment befinden Sie sich im CCP, so daß auch im Ernstfall nichts verloren wäre. Wenn das aber in einem Anwenderprogramm geschieht, so wird durch die ERROR-Meldung oder auch mit einem Warmboot immer das Programm verlassen, und den Weg zurück werden Sie nicht mehr finden.

ALLGEMEINES ÜBER CP/M PLUS

Jetzt wollen wir zu CP/M Plus übergehen und uns zunächst das ansehen, was dort gegenüber der Version 2.2 verbessert ist. Eine der wesentlichen Verbesserungen ist darin zu sehen, daß CP/M Plus zwischen physikalischen und logischen Laufwerken unterscheidet.
Dadurch ist es möglich, daß einem vorhandenen physikalischen Laufwerk (zum Beispiel beim CPC 6128) mehrere (zwei beim CPC) logische Laufwerke zugeordnet werden. Machen Sie gleich einen Versuch: Starten Sie CP/M Plus mit einem Laufwerk. Sie erhalten eine entsprechende "Bootmeldung" und in der unteren (Status-)Zeile die Meldung, daß Laufwerk A aktiv ist. Wenn Sie zwei Laufwerke angeschlossen haben, beim Start von CP/M Plus sich aber keine Diskette im Laufwerk B befindet, so erhalten Sie die gleiche Meldung: Das System erkennt das zweite Laufwerk nicht. Achten Sie also immer darauf, daß sich bereits beim Start (aber nach dem Einschalten des Laufwerks!) eine Diskette in B befindet.
Sie können sich auch davon überzeugen, daß dies bei CP/M 2.2 nicht erforderlich ist. Wenn Sie noch kein Zweitlaufwerk besitzen, sich aber eines anschaffen wollen, so wählen Sie doch ein 5 1/4-Zoll-Laufwerk mit 2 x 80 Spuren. Bei CP/M Plus befindet sich das Disk-BIOS im RAM (siehe "CP/M-Beitrag") und läßt sich daher so patchen, daß das Laufwerk vom System akzeptiert wird. Dann haben Sie endlich eine ausreichende Diskettenkapazität.
Wählen Sie nun Laufwerk B durch Eingabe von "B:". Während das bei CP/M 2.2 zu einem Fiasko führte, fordert Sie jetzt das System in der Statuszeile auf, die Diskette für Laufwerk B einzufügen und dies durch die Betätigung einer beliebigen Taste zu quittieren. Damit steht Ihnen auf dem gleichen **physikalischen** Laufwerk das **logische** Laufwerk B zur Verfügung. Das funktioniert natürlich genauso "programmgesteuert", das heißt, wenn ein beliebiges CP/M-Programm den Zugriff auf Laufwerk B verlangt, meldet sich das System in gleicher Weise. Was beim CP/M 2.2 unmöglich war, geht hier absolut einwandfrei – wenn auch vielleicht ein wenig holperig. Wir werden das später bei PIP noch exerzieren. Eine zweite Verbesserung von CP/M Plus, daß es nämlich einen Diskettenwechsel ohne ERROR akzeptiert, kann man aufgrund des oben geschilderten vermuten. Wir werden uns später davon noch überzeugen.

CP/M PLUS RESIDENTE COMMANDS

Jetzt wollen wir die **residenten** Commands von CP/M Plus betrachten. Es sind fast dieselben wie bei CP/M 2.2, es fehlt nur SAVE (dieses Command ist hier **transient**). Bei CP/M Plus gibt es aber für die **residenten** Commands DIR ERA REN und TYPE **transiente** Ergänzungen. Wir probieren zunächst einmal das residente **DIR**. Das Ergebnis ist praktisch wie bei 2.2. Es gibt einen Unterschied, den Sie aber bei den drei Disketten – mangels Masse – kaum bemerken werden. Wenn der Bildschirm voll ist, hält die Ausgabe an, und der CPC fordert Sie auf, RETURN zu drücken, wenn Sie alles gelesen haben. Bei **TYPE** können Sie diesen Effekt besser beobachten. Daß Sie auch hier die Bildschirmausgabe durch vorherige Eingabe von CTRL-P auf dem Drucker protokollieren können, wissen Sie bereits. Auch der "Beep" wird Sie hoffentlich nicht mehr erschrecken. Vergessen Sie aber nicht das abschließende Ausschalten – ebenfalls mit CTRL-P. Vielleicht stört Sie die immer wiederkehrende "RETURN"-Aufforderung bei der Druckerausgabe. Hier hilft das **transiente TYPE**, bei dem Sie die "Option NOPAGE" eingeben können. Das sieht dann so aus **TYPE dateiname [NOPAGE]**. Die Option muß in **eckigen** Klammern stehen! Das kann sich auf Bildschirm (und Drucker) als deutsche Umlaute darstellen, wenn Sie den deutschen Zeichensatz installiert haben (siehe unten). Zur Ausführung dieses transienten Commands brauchen Sie natürlich – wie bei allen solchen Commands – das entsprechende Programm, hier also TYPE.COM. Was tun Sie, wenn zwar die auszugebende Datei auf der Diskette steht, nicht aber TYPE.COM? Geben Sie ein **B:TYPE A:dateiname [NOPAGE]**. Das System fordert Sie auf, die Diskette B einzulegen, auf der natürlich TYPE.COM stehen muß. Dies wird geladen, dann werden Sie aufgefordert, die Diskette A einzulegen, und schon läuft die Ausgabe auf Bildschirm und Drucker wie geschmiert. Moment, Sie wollen jetzt doch anhalten? Geben Sie einfach **CTRL-S** ein – wie früher bei 2.2 – und wenn es weiter gehen soll, hilft **CTRL-Q** (aber hier **keine** andere Taste). Haben Sie das Prinzip verstanden? Wir probieren es gleich noch einmal bei **DIR**, denn dort gibt es eine besonders interessante Option, nämlich [FULL] (die Option muß wieder in eckigen Klammern stehen.). Sie wollen also das Directory der eingelegten Diskette, auf der sich aber **nicht** das Programm DIR.COM befindet. Also geben Sie ein **B:DIR A [FULL]** Nach Aufforderung legen Sie die Diskette B (mit Programm DIR.COM) ein und dann – wieder nach Aufforderung – erneut die Diskette A. *Sie wollen es etwas anders machen? Bitte sehr!*

Sie starten CP/M mit einer Systemdiskette, auf der alle gewünschten transienten Systemprogramme enthalten sind. Das ist dann Laufwerk A. Jetzt geben Sie ein **DIR B [FULL]**. Nach Aufforderung legen Sie dann "Ihre" Diskette als B ein und erhalten die Directory-Ausgabe. Natürlich können Sie wieder durch vorherige Eingabe von CTRL-P auf dem Drucker mitprotokollieren. Die Ausgabe wird Sie an die Auflistung erinnern, die wir bei CP/M 2.2 mit STAT erhielten. Sie sehen Angaben zur Größe jeder

Datei. Zunächst der auf der Diskette belegte Platz in kByte. Da beim 3"-Format die Blockgröße 1 kByte beträgt, ist dies auch die Anzahl der belegten Blöcke.

Aus dem "CP/M-Beitrag" wissen Sie, daß die Datei immer eine ganze Zahl von Blöcken belegt, auch wenn sie etwas kleiner ist. Die tatsächliche Dateigröße ist ebenfalls angegeben und zwar in "records" (1 record = 128 bytes, so daß 8 records = 1 kByte.) Beim VORTEX-Format ist die Blockgröße 1 kByte. Dadurch kann sich hier für die gleiche Datei ein größerer Platzbedarf ergeben. Eine Datei von 15 records zum Beispiel belegt unter AMSDOS 2 Blöcke gleich 2 kByte, unter VORTEX aber 1 Block gleich 4 kByte. Weiter sind in der DIR-Ausgabe bei jeder Datei "Attribute" angegeben. Diese Attribute werden mit dem transienten Command **SET** gesetzt (und gelöscht), wie wir noch sehen werden. "RW" ist eine Datei, die gelesen und geschrieben werden kann, also der Normalfall. "RO" (Read Only) ist eine schreibgeschützte Datei. Weiter sind alle Dateien entweder "Dir"-(Normalfall) oder "Sys"-Dateien. Die letzteren spielen eine Rolle in Verbindung mit dem **USER**-Command (siehe unten). JOYCE-Benutzer kennen "Sys"-Dateien vom LOCOSCRIPT her. Mit dem residenten **DIR** werden diese Dateien nicht angezeigt. Es erfolgt lediglich der Hinweis, daß welche vorhanden sind.

Es gibt aber das residente Command **DIRSYS** – oder auch **DIRS** – mit dem diese Dateien aufgelistet werden. Bitte beachten Sie: "Sys"-Dateien haben das "Attribut" Sys, welches mit **SET** gesetzt wird. Dem **Dateinamen** sieht man das nicht an. Sie müssen also nicht vom **Typ .SYS** sein! Die weiteren Ausgaben im **DIR [FULL]**-Command wollen wir hier nicht besprechen, Sie erfahren darüber mehr in der Fachliteratur. Wenn Sie nur an der Platzbelegung der Dateien interessiert sind, so können Sie auch die Option **[SIZE]** bei dem Command **DIR** verwenden. Auch das Command **ERA** hat eine transiente Ergänzung, die manchmal ganz nützlich sein kann. Das **residente** Command funktioniert ähnlich wie unter CP/M 2.2, lediglich bei der Verwendung von "Wildcards" fragt es zur Vorsicht immer nach.

Manchmal werden Sie aber mit den "Wildcards" nicht auskommen. Wir verweisen auf unser oben gezeigtes Beispiel mit den Dateien BILANZnn (nn = Jahreszahl). Sie wollen alle "Siebziger"-Bilanzen löschen, nur nicht "78". Mit der Option **[CONFIRM]** (abgekürzt auch **[C]**) geht das ganz einfach. (Sie müssen jetzt natürlich wieder das transiente Programm ERASE.COM zur Verfügung haben.) Sie rufen auf *ERA BILANZ7? [CONFIRM]*. Dann fragt das System bei jeder "Siebziger"-Bilanz, ob sie gelöscht werden soll. Einige Worte wollen wir noch zu dem residenten Command **USER** sagen.

Sie wissen vielleicht, daß bei MS-DOS auf der Diskette neben dem "Haupt"-Directory auch "Unter"-Directories definiert werden können. So läßt sich bei einer Vielzahl von Dateien leichter Ordnung halten. Bei CP/M kann man mit **USER** etwas ähnliches erreichen. Sie können USER-Bereiche von 0 bis 15 auf der Diskette anlegen und in jedem Bereich Dateien einer bestimmten Sachgruppe speichern.

Sie geben zum Beispiel ein: **USER 4**, und dann steht Ihnen der Bereich 4 auf der Diskette zur Verfügung. Sie sehen das an der CCP-Meldung. Dort steht nämlich die USER-Nummer vor dem Laufwerk, also "4A>". Jetzt können Sie Dateien in diesem Speicherbereich (und auch – mit einer Ausnahme – **nur** in diesem Bereich) schreiben und lesen. Die Ausnahme bilden Dateien im Bereich 0, welche das Attribut "Sys" haben. Auf diese können Sie von **jedem** USER-Bereich aus zugreifen. Das ist der eigentliche Sinn der "Sys"-Dateien.

Mit PIP können Sie übrigens zwischen den einzelnen USER-Bereichen kopieren (siehe unten). Es gibt aber noch einen anderen Weg, Dateien von einem USER-Bereich in einen anderen zu bringen: Wenn Sie sich das Directory einer Diskette mit mehreren USER-Bereichen mit einem Diskettenmonitor ansehen, so bemerken Sie, daß das erste Byte eines Eintrages die USER-Nummer angibt (Hex 0 bis F). Sie können nun diese Einträge mit Hilfe des Monitors ändern und dadurch die Datei in den gewünschten Bereich "befördern".

CP/M PLUS TRANSIENTE COMMANDS

Jetzt wollen wir uns den "eigentlichen" **transienten** Commands von CP/M Plus zuwenden. Das Command **STAT** existiert hier nicht. Sie wissen, daß damit unter CP/M 2.2 der freie Diskettenplatz und die Dateibelegung angezeigt werden (STAT tut noch einiges mehr, was wir nicht besprochen haben). STAT benutzt einen BDOS-Aufruf, der unter CP/M Plus nicht funktioniert. Daher wird es hier durch zwei andere transiente Commands ersetzt: **SHOW** für den freien Diskettenplatz und **DIR** für die Dateibelegung. **DIR** haben wir schon kennengelernt. **SHOW** funktioniert genauso wie **STAT**. Probieren Sie es aus. Wenn Sie das transiente Programm **SHOW.COM** nicht auf der Diskette haben, so können Sie sich nach dem oben angegebenen Muster helfen. Es gibt noch verschiedene Optionen, die wir hier aber nicht erläutern können. *Ein Hinweis noch für Benutzer von TURBO-PASCAL.* Der DIR-Befehl gibt dort die freie Diskettenkapazität an. Unter CP/M Plus stimmt das aber nicht, weil die gleiche BDOS-Funktion verwendet wird wie bei STAT. Mit einem – von Borland angegebenen – Patch kann man diesen Fehler beseitigen. Mit **PIP.COM** können wir unter CP/M Plus auch mit einem physikalischen Laufwerk bestens arbeiten, denn wir haben ja zwei logische Laufwerke. Die Handhabung ist einfach. Am besten rufen Sie zunächst einfach immer **PIP** auf. Es meldet sich die CP/M-Plus-Version mit dem Prompt "*". Dann müssen Sie eingeben nach dem Schema:

ziellaufw:zieldatei=quelllaufw: quelldatei

Also zum Beispiel **B:DATEIBB.TYP= A:DATEIAA.TYP** . Wenn der Dateiname derselbe bleibt, so wird daraus einfacher:

ziellaufw:=quelllaufw:dateiname
(**B:=A:DATEI00.TYP**).

Bei Dateiname können Sie "Wildcards" verwenden und dadurch ganze Dateigruppen kopieren, zum Beispiel: **B:=A:DATEI*.TYP** .

Es werden die vorhandenen Dateien DATEI00.TYP DATEI01.TYP und DATEI02.TYP kopiert. Wenn ein "Kopierauftrag" beendet ist, so erscheint wieder der Prompt, und Sie können einen weiteren "Auftrag" eingeben. Nachdem alles erledigt ist, geben Sie einfach RETURN ein, und PIP wird damit beendet. Wenn Sie noch nicht mit PIP gearbeitet haben, so probieren Sie es gleich einmal aus, um zu erleben, wie das System Sie immer wieder zum Diskettenwechsel auffordert und so die beiden logischen Laufwerke "managt". Es gibt eine Reihe von Optionen, die wieder in eckige Klammern gesetzt werden müssen.

[V] (= Verify) bedeutet, daß nach dem Kopieren jeder Datei Quelle und Ziel auf Gleichheit überprüft werden. Sie sind damit ganz auf der sicheren Seite, müssen aber entsprechend lange warten. Erfahrungsgemäß ist die Verwendung im allgemeinen **nicht** erforderlich.

[R] bedeutet, daß "Sys"-Dateien kopiert werden, wobei auch die Zieldatei dieses Attribut bekommt (ohne diese Option werden "Sys"-Dateien **nicht** kopiert). Diese beiden Optionen werden an den Schluß des Kopierauftrags (hinter die Quellendatei) gesetzt und gelten dann für Quelle und Ziel. Zum Beispiel:
B:=A:DATEI00.TYP[V]

Die Option **[Gnummer]** (Nummer gleich 0 bis 15) dient zum Kopieren von Dateien von einem USER-Bereich zum anderen. Also:
A:[G2]=B:FDATEN.TST[G1]
kopiert die Datei FDATEN.TST aus dem USER-Bereich 2 der Diskette im Laufwerk A (unter demselben Namen) in den USER-Bereich 1 einer anderen Diskette im Laufwerk A.
Es geht aber auch das folgende:
A:[G2]=A:FDATEN.TST [G1] .
Hier wird **auf der gleichen Diskette** die Datei vom USER-Bereich 1 in den USER-Bereich 2 kopiert. Die Datei ist dann in beiden USER-Bereichen vorhanden, was keineswegs stört. So kann man nachträglich auf einer Diskette USER-Bereiche einrichten. Das hat allerdings eine kleinen Haken. Die doppelt vorhandene Datei braucht auch doppelten Platz. Hinterher kann man zwar dann die Datei in einem USER-Bereich löschen, aber zunächst muß der Platz da sein. *PIP* überschreibt eine bereits vorhandene Datei gleichen Namens, ohne daß eine Backup-Datei erstellt wird.
Wir nehmen an, daß sich auf der Diskette in A bereits eine Datei TEST.TST befindet und daß mit PIP eine neue Version dieser Datei von B nach A kopiert werden soll. Dann geschieht folgendes: PIP richtet auf A eine Datei TEST.**$$$** ein und kopiert dorthin die Version von B. Dann wird die alte Datei TEST.TST auf A gelöscht und die Datei TEST.**$$$** in TEST.TST umbenannt. (Ein Programm, welches ein Backup erstellt, wird die alte Datei nicht löschen, sondern in TEST.BAK umbenennen.)
Wenn nun auf A für die Datei TEST.**$$$** kein Platz mehr ist, so streikt PIP. Sie können sich dann helfen, indem Sie vorher die alte Version von TEST.TST auf A löschen. Statt des Programms DDT gibt es unter CP/M Plus das Programm **SID.COM** (Symbolic Instruction Decoder). Sie wissen, daß man unter CP/M 2.2 beim Arbeiten mit DDT unbedingt das residente SAVE benötigt, um das Ergebnis eines Patches abzuspeichern. Unter CP/M Plus ist SAVE ein transientes Programm, bei dessen Handhabung man leicht Fehler machen kann. Glücklicherweise ist aber **SID** ein Programm, mit dem wir nicht nur die Patches im RAM durchführen, sondern anschließend gleich auf der Diskette abspeichern können, so daß SAVE nicht benötigt wird. Zur Einübung wollen wir die gleichen Beispiele unter CP/M Plus exerzieren, die wir oben unter CP/M 2.2 ausgeführt haben (Listing la und lb). Wir müssen lediglich den Steuercode für das Bildschirmlöschen ändern. Unter CP/M Plus muß es heißen 1B 48 1B 45 (ESC H ESCE). An diesem Beispiel erkennen Sie auch den Grund, weshalb CP/M-Programme, die doch auf allen CP/M-Rechnern laufen sollen, unter Umständen erst "installiert" werden müssen. Die Steuerungen von Bildschirm und Drucker sind doch zum Teil sehr unterschiedlich, so daß hier entsprechende Patches erforderlich sind, die meist mit "Installationsprogrammen" ausgeführt werden (zum Beispiel WORDSTAR, MULTIPLAN, dBASE, TURBO PASCAL usw.). Abbildung 9 zeigt das Protokoll der "SID-Sitzung". Nach dem Aufruf von SID läuft zunächst alles genauso ab wie bei DDT (in Abbildung 7), denn die Commands von SID entsprechen weitgehend denen von DDT. Lediglich der Abschluß ist jetzt anders. Wenn Sie durch Eingabe eines Punktes den Modus für das S-Command verlassen haben, dürfen Sie SID **nicht** beenden, sondern müssen zuerst das im RAM stehende Programm speichern. Das geschieht mit dem Command W. Es lautet:
Wlaufw:dateiname.typ,anfadr,endadr
Wenn das Speichern auf dem angemeldeten Laufwerk geschehen soll, so brauchen Sie es nicht anzugeben. Wir geben also ein: **W TEXTAUS.COM,0100,011B**, so wie sich die Adressen bei der Eingabe mit dem S-Command ergeben. Das Umrechnen in Pages können wir uns hier sparen, SID teilt uns mit, daß es etwas mehr abgespeichert hat, nämlich 1 record (von 0100 bis 017F). Danach können Sie SID mit CTRL-C verlassen. Sie können auch – und jetzt gefahrlos, denn das Programm ist gespeichert – dieses mit **G0100** von SID aus starten. Das Patching entsprechend dem Protokoll in Abbildung 10 wird Ihnen ohne weiteres verständlich sein.
Wir wollen uns jetzt noch davon überzeugen, daß CP/M Plus einen Diskettenwechsel ohne Murren akzeptiert. Wir verfahren ähnlich wie bei CP/M 2.2, benutzen aber jetzt das Programm **SID** zum Speichern. Rufen Sie also zunächst **SID** auf, Sie wissen, daß Sie mit dem **W** Command-Daten aus dem RAM auf der Diskette speichern können. Jetzt merken Sie aber wieder – wie oben bei CP/M 2.2 –, daß auf dieser Diskette kein Platz mehr ist, und legen "heimlich" eine neue ein. Dann geben Sie ein **W TEST.TST 0100,03FF**. Ergebnis: Die Daten werden anstandslos auf der neuen Diskette gespeichert. Glauben Sie aber nicht, daß CP/M geschlafen hat. Es hat die neue Diskette sehr wohl erkannt und von sich aus neu "eingelogged". Damit ist alles wieder in Ordnung. Das transiente Command **SET** (Programm SET.COM) haben wir schon erwähnt. Man kann damit verschiedene "Datei-Attribute" setzen. Wenn Sie eine Datei als "Sys" definieren wollen, so rufen Sie auf **SET laufw:dateiname.typ[SYS]** . Wollen Sie das wieder rückgängig machen, so geben Sie ein **SET laufw:dateiname.typ[DIR]** . Sie können auch eine Datei als "schreibgeschützt" deklarieren – was vielleicht manchmal ganz nützlich ist. Sie geben dann ein **SET laufw:dateiname.typ[RO]**.
Mit **SET laufw:dateiname.typ[RW]** machen Sie das wieder rückgängig. Erwähnen wollen wir das Command **SETDEF**. Wenn Sie ein transientes Programm – also eine .COM-Datei – zur Ausführung aufrufen, so sucht der CCP normalerweise danach nur auf der Diskette im angemeldeten Laufwerk. (Es sei denn, Sie haben durch Eingabe von "laufwerk:" vor dem Dateinamen ein anderes Laufwerk benannt, auf dem gesucht werden soll.) Mit **SETDEF** können Sie in einem "Suchpfad" alle Laufwerke angeben, in denen das System nach dem Programm suchen soll. Bei den zwei Laufwerken des CPC ist das nicht besonders wichtig. Eher vielleicht beim Joyce, wo die RAMDISK M noch dazu kommt. Wenn Sie MS-DOS kennen, so werden Sie die Parallele zu PATH erkennen. Nützlich sind noch die transienten Commands **SUBMIT** und **PROFILE** . Mit **SUBMIT** können Sie eine "SUBMIT-Datei" abarbeiten lassen. Eine solche Datei ist eine "Batchdatei" ("Stapeldatei"), die **.SUB** als Typbezeichnung haben muß. Sie kann mit einem Texteditor erstellt werden und enthält eine Reihe von Commands, die Sie sonst dem CCP eingeben könnten. Sie schreiben zum Beispiel mit Wordstar (im "N"-Modus!) den Text:

```
DIR
WS
```

und speichern ihn als Datei **ANFANG.SUB**. Wenn Sie jetzt eingeben **SUBMIT ANFANG** , so wird erst das Directory ausgegeben und dann Wordstar gestartet; genauso, als wenn Sie die Commands von Hand eingegeben hätten. Daher der Name "Stapeldatei", weil die "gestapelten" Commands der Reihe nach abgearbeitet werden. Besonders interessant ist es, wenn die Commands in einer SUBMIT-Datei mit dem Namen **PROFILE.SUB** stehen, weil diese Datei beim Start des Rechners *automatisch* abgearbeitet wird. Beim Booten sucht das System im Directory nach einer Datei **PROFILE.SUB**. Ist sie vorhanden, so wird sie vom System mit **SUBMIT PROFILE** aufgerufen. Nützlich ist es, wenn in einer solchen PROFILE-Datei Commands stehen, mit denen eine gewünschte "Grundeinstellung" des Rechners erfolgt. Beim CPC gibt es im we-

```
A> SID TEXTAUS.COM
CP/M 3 SID - Version 3.0
NEXT MSZE  PC  END
0180 0180 0100 D2FF
# D0100,017F
0100: 11 0B 01 0E 09 CD 05 00 C3 00 00 1B 48 1B 45 47 ............H.EG
0110: 75 74 65 6E 20 4D 6F 72 67 65 6E 24 00 00 00 00 uten Morgen$....
0120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0150: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0160: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
0170: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
# S0115
0115 4D  41
0116 6F  62
0117 72  65
0118 67  6E
0119 65  64
011A 6E  20
011B 24
011C 00  .
# W TEXTAUS.COM,0100,011C
0001h record(s) written.
# ^C
A>
```

Abb.10: Protokoll des Patches von TEXTAUS.COM unter CP/M Plus

sentlichen zwei Dateien, die gerätespezifische transiente Commands darstellen, nämlich **SETKEYS.COM** und **LANGUAGE.COM** . Beim CPC sind die "Tastencodes" für die Bedürfnisse von BASIC ausgebildet. (Außerdem hat er eine "amerikanische Tastatur" "QWERTY".) Für CP/M-Programme (zum Beispiel Textverarbeitung) ist es aber besser, wenn die Tasten entsprechend anders codiert sind, zum Beispiel die Pfeiltasten mit den Cursor-Steuercodes, die ESCAPE-Taste mit ASCII ESC (1B) usw.
In der Datei **KEYS.WP** ist etwas derartiges programmiert, und wenn Sie **SETKEYS KEYS.WP** aufrufen, so wird Ihre Tastatur für die Textverarbeitung "umcodiert". Probieren Sie das einmal aus, und probieren Sie auch **SETKEYS KEYS.CCP.** In der Bedienungsanleitung finden Sie Hinweise, wie Sie sich eine KEYS-Datei nach Ihren Wünschen erstellen können. Weiter sind bei CP/M Plus die "Matrizen", mit deren Hilfe die Darstellung der ASCII-Zeichen auf dem Bildschirm erfolgt, im RAM gespeichert (Bei CP/M 2.2 befinden sie sich im ROM!). Das heißt, bei CP/M Plus können die Zeichen leicht geändert werden. Dazu dient die Datei **LANGUAGE.COM**. "Von Hause aus" gilt der "amerikanische" ASCII-Zeichensatz (mit eckigen und geschweiften Klammern). Mit dem Aufruf von **LANGUAGE 2** erhalten Sie den "deutschen" Zeichensatz. Jetzt wird bei der Betätigung der Taste "[" der Buchstabe "Ä" und bei zusätzlicher Betätigung von **SHIFT** (= Taste "[") der Buchstabe "ä" ausgegeben. (Durch Eingabe von **LANGUAGE 0** erhalten Sie wieder den amerikanischen Zeichensatz.)
Wenn es Sie stört, daß bei den Umlauten die Kleinbuchstaben mit der SHIFT-Taste erzeugt werden müssen, so können Sie das mit einer geeigneten KEYS-Datei ändern.
Derartige Commands eignen sich nun besonders für eine Datei PROFILE.SUB. Sie könnte etwa folgendermaßen aussehen:

SETKEYS KEYS.WP
LANGUAGE 2
CURBLK30
WS

Auf der Systemdiskette müssen außer dieser Datei PROFILE.SUB noch folgende Dateien vorhanden sein:

SETKEYS.COM
KEYS.WP
LANGUAGE.COM
CURBLK30.COM
SUBMIT.COM
WS.COM

Beim Start des Rechners wird dann die Tastatur des Rechners für Wordstar eingerichtet, die Bildschirmdarstellung für den deutschen Zeichensatz, ein "blinkender" Cursor, wird erzeugt, und Wordstar wird gestartet – ohne daß Sie eine zusätzliche Taste betätigen müssen. Bei einem Farbmonitor können Sie noch mit dem Programm PALETTE.COM Hintergrund- und Schriftfarbe nach Ihren Wünschen wählen.

Einige weitere transiente Commands von CP/M Plus wollen wir noch kurz erwähnen: **GET.COM** ist ein Programm, mit dem man Eingaben für den CCP, die ja normalerweise "von Hand" über die Tastatur kommen, aus einer Datei "holen" kann. Auf diese Weise kann man zum Beispiel in einer SUBMIT-Datei beliebige CCP-Eingaben unterbringen. **PUT.COM** leitet die Bildschirmausgabe in eine Datei um. Unter CP/M Plus stehen gleich drei Assembler (mit Hilfsprogrammen) zur Verfügung: ASM, MAC und RMAC, alle allerdings nur für den "8080 Code". Das Programm DATE.COM dient zum Stellen und Anzeigen der "inneren" Uhr des CPC (mit Datumsanzeige). Leider geht diese Uhr nur, wenn der Rechner eingeschaltet ist (keine eingebaute Batteriepufferung). Daher muß sie jedesmal neu gestellt werden.

Damit soll unser Streifzug durch die residenten und transienten CP/M-Commands beendet sein. Den Einstieg in die CP/M-Welt werden Sie mühelos schaffen, und mit der angegebenen Literatur können Sie sich sogar bald zu einem "CP/M-Freak" entwickeln.

(Hans-Georg Sanner/cd)

Literaturhinweise:

Jürgen Hückstädt, "CP/M 2.2 Anwenderhandbuch CPC 464/664/6128"
Markt&Technik 1985
Jürgen Hückstädt, "CP/M Plus Anwenderhandbuch CPC 6128/Joyce"
Markt&Technik 1986
Als Lehrbücher und Nachschlagewerke für die CP/M-Commands geeignet. Im zweiten Werk sind dem Verfasser bei der allgemeinen Systembeschreibung einige kleine Fehler unterlaufen.

Jörg Schieb, "CP/M"
DATA BECKER Führer 1986,
Bernd Grohmann, "Schneider CPC"
DATA BECKER Führer 1986.
Als Nachschlagewerke sehr gut geeignet. Das zweite Werk enthält zusätzlich Information über CP/M-Software und BASIC-Befehle. Ferner allgemeine System-Information über CP/M und die CPC Hardware.

Rodnay Zaks, "CP/M Handbuch"
Sybex, 2.Aufl. 1984
Eine allgemeine Einführung in CP/M 2.2.

Helmut Tischer, "Programmentwicklung unter CP/M 2.2 auf dem CPC 464/664"
Markt&Technik 1986
Eine sehr tiefgehende Behandlung von CP/M 2.2. Für Systemprogrammierer bestens geeignet.

Klaus Kämpf, "CP/M 2.2 Assembler-Listing"
Röckrath Microcomputer Aachen 1985
Listings von CCP und BDOS mit vielen weiteren Hinweisen. Wichtig für interessierte Systemprogrammierer.

Werner Schmuck, Manfred Schuler, Willi Wagemuth "Einsteigen in CP/M"
c't 1985 Heft 7, 8, 9, 10 (CP/M 2.2)
c't 1985 Heft 11, 12 1986 Heft 1, 2 (CP/M Plus)
Diese sehr ausführliche und weitgehende Aufsatzreihe kann weitgehend die Original-Dokumentation von DIGITAL RESEARCH ersetzen.
Diese Dokumentation – in englischer Sprache – ist teilweise als Nachdruck bei Markt&Technik erschienen.

M. Anton, "Im Herzen von CP/M Plus - XBIOS zerlegt"
JOYCE Sonderheft 1/1987
Dieser Aufsatz bringt sehr viel Wissenswertes über CP/M Plus – auch für den CPC 6128.

Martin Kotulla,
"Programmieren unter CP/M Plus"
"CP/M Plus ist einfach mehr".
"Am Pulsschlag der CPU"
Sonderheft 10 (5. Schneider Sonderheft)
HAPPY COMPUTER.
Es werden die BDOS- und BIOS-Funktionen beschrieben. Dem Verfasser sind leider einige kleine Fehler unterlaufen.

FAst BAsic COMpiler



BASIC-Compiler für CPC 464/664/6128

Der Turbo-Antrieb für Ihre BASIC-Programme!

Haben auch Sie sich schon immer gewünscht, daß Ihre selbstgeschriebenen BASIC-Programme schneller laufen? Mit dem BASIC-Compiler von DMV ist das nun kein Problem mehr, denn

- der Compiler hat den vollen Sprachumfang des BASIC 1.1 (CPC664/6128).
- das compilierte Programm ist auf jedem CPC lauffähig.
- unterstützt Integer- und Fließkomma-Arithmetik.
- ist kompatibel zu Vortex-Peripherie inkl. Nutzen der RAM-Disk.
- Programme, die spezielle BASIC-1.1.-Befehle beinhalten, sind auch auf dem CPC464 lauffähig (außer FILL und MASK).
- der Compiler arbeitet unter CP/M, das heißt, alle CP/M-Dienstprogramme können genutzt werden.
- bis 17 kByte Quellcode können problemlos compiliert werden.
- einzelne Programmteile können ebenfalls compiliert werden (z.B. wichtig bei Nachladeprogrammen).
- die ausführliche deutsche Bedienungsanleitung macht Sie auf einfache Weise mit dem Umgang des Compilers vertraut.
- viele Beispielprogramme veranschaulichen die Arbeitsweise des Compilers und zeigen die Geschwindigkeitsvorteile auf.
- das Programm ist in 100% Maschinencode geschrieben.

Der BASIC-Compiler ist nur auf 3"-Diskette erhältlich.

Best.-Nr.: 209 **Preis 49,– DM***

Bitte benutzen Sie die Bestellkarte.

Möchten Sie gerne in Assembler programmieren ?

CPC Assembler Software und Trainingsbuch

Das Buch:

führt den Anfänger schrittweise in die Programmierung des Z80 ein. Dabei werden Vorkenntnisse nicht vorausgesetzt. Nach der Lektüre des Buches sind Sie mit dem Befehlssatz des Prozessors wie auch mit den Adressierungsarten vertraut. Anhand einer Fülle von Programmbeispielen, die speziell auf den CPC 464 zugeschnitten sind, lernen Sie, nützliche Routinen in Maschinensprache zu entwickeln, die auf Ihrem Rechner sofort lauffähig sind. Dazu erfahren Sie, wie Sie die im ROM des Schneider CPC vorhandenen Hilfsroutinen sinnvoll für die Lösung eigener Probleme einsetzen können. Darüber hinaus lernen Sie einige spezielle Programme zur Erweiterung des Betriebssystems mit leistungsfähigen grafischen Befehlen kennen, die Sie in BASIC verwenden können, z.B. TRIANGEL, BOX und CIRCLE. Ein umfangreicher Anhang mit Erläuterungen des Befehlssatzes, des Assemblers sowie einer Vielzahl kommentierter Einsprungadressen des Betriebssystem-ROMs rundet das Buch ab.



Die Software:

besteht aus einem menügesteuerten Z80 - Assembler.
Mit seiner Hilfe können Sie nicht nur die im Buch erläuterten Beispielprogramme editieren und in Maschinensprache übersetzen, sondern auch selbst entworfene Programme entwickeln und testen. Für Ihre Arbeit stehen Ihnen Funktionen zur Verfügung wie **Einfügen – Löschen – Ändern von Programmzeilen – Abspeichern und Laden von Programmen auf Datenträgern wie Kassette und Diskette – Ausgabe von Listings auf Bildschirm und Drucker – Zahlenkonvertierung von der Menü-Ebene aus – Durchführung von Zwischenrechnungen mit binären, hexadezimalen und dezimalen Zahlen**. Weiterhin erhalten Sie ein Trainingsprogramm, mit dessen Hilfe Sie den Umgang mit verschiedenen Zahlenformaten vertiefen können, und Programme zur Demonstration der grafischen Erweiterungen des Betriebssystems.

Buch mit Kassette Best.-Nr.: 446 **Preis 39,– DM***

Buch mit Diskette Best.-Nr.: 447 **Preis 49,– DM***

*Unabhängig von der Anzahl der bestellten Programme berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag

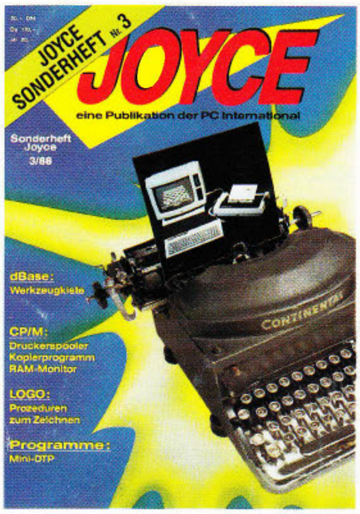
SENSATIONELL

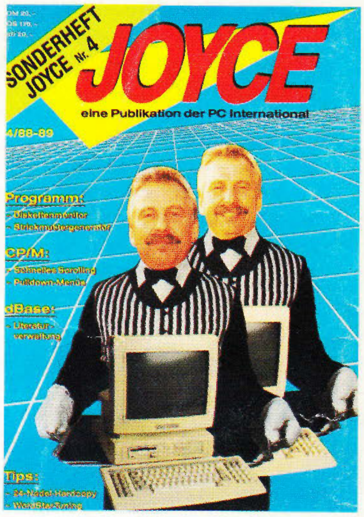
JOYCE

begrenzter Vorrat



Kühles Wetter, heiße Preise





Sonderhefte Joyce Nr. 2, 3, 4

als Paket
für sagenhafte **29,– DM***

DMV braucht Platz für neue Ideen –
darum müssen wir unsere Lager räumen.

Allen JOYCE-Besitzern machen wir deshalb ein einmaliges Angebot:
JOYCE Sonderhefte 2, 3 und 4 sind ab sofort im Kombi-Pack zum Preis von 29,– DM beim Verlag zu beziehen.
– Sie sparen mehr als 50% gegenüber dem Einzelpreis!

JOYCE Sonderhefte sind Sonderpublikationen der PC AMSTRAD International und bieten jeweils auf 120 Seiten ausschließlich erstmalig veröffentlichte Beiträge, Tips und Tricks zu PCW 8256, 8512 und 9512.

Aus dem Inhalt
JOYCE Sonderheft 2:
Adreßverwaltung
Archivprogramm
(Video-oder Literaturverwaltung)
PASCAL-Compiler in BASIC
Suburbia (Spiel ähnlich Monopoly (R))
Turbo-PASCAL-Grafikroutinen ohne GSX
Komfortable Balkengrafik
JOYCE-Zweitlaufwerk selbst anschließen
3-D-Plotter
Etikettendruckprogramm
Ordnung auf der Diskette mit LocoScript
Funktionstasten selbst belegen
Spaltensatz unter LocoScript
dBase-Handbuch selbst ausdrucken
LOGO- Funktionenzeichner

Aus dem Inhalt
JOYCE Sonderheft 3:
Vokabeltrainer
RAM-Monitor – Speicherinhalte verändern
Memory-Spiel
Mini-DTP-Programm
Drucker-Spooler unter CP/M
Disketten-Kopierprogramm bis 43 Spuren
Reset ohne Datenverlust
Grafik auf dem JOYCE-Drucker
Tastaturbelegung unter CP/M und LocoScript ändern
Reset ohne Datenverlust
Super-Werkzeugkiste dBase
Grafikutilities für LOGO

Aus dem Inhalt:
JOYCE Sonderheft 4:
Strickmustergenerator
WordStar-Verbesserungen
Bundesligasimulator
Super Reaktionsspiel
FILEMANAGER, Pulldown-Menüs
Stichwortverzeichnis,
Astrologieprogramm
Diskettenmonitor
Hauptstädte raten in LOGO
Statuszeile für dBase und Basic
Hardcopyroutine für 24-Nadler
LOGO macht Schachteln
dBase-Literaturverwaltung
Universelles Werkzeug zur Veränderung von dBase-Dateien

Und dazu die Databoxen aller Sonderhefte im Paket !

---

5 Disketten Joyce-Power für traumhafte 79,– DM*

*** Unabhängig von der Anzahl der bestellten Artikel berechnen wir für das Inland 4,– DM bzw. für das Ausland 6,– DM Porto und Verpackung.**

Bitte benutzen Sie den Bestellcoupon auf Seite 7.

DMV-Verlag · Postfach 250 · 3440 Eschwege

DMV Daten- und Medienverlag